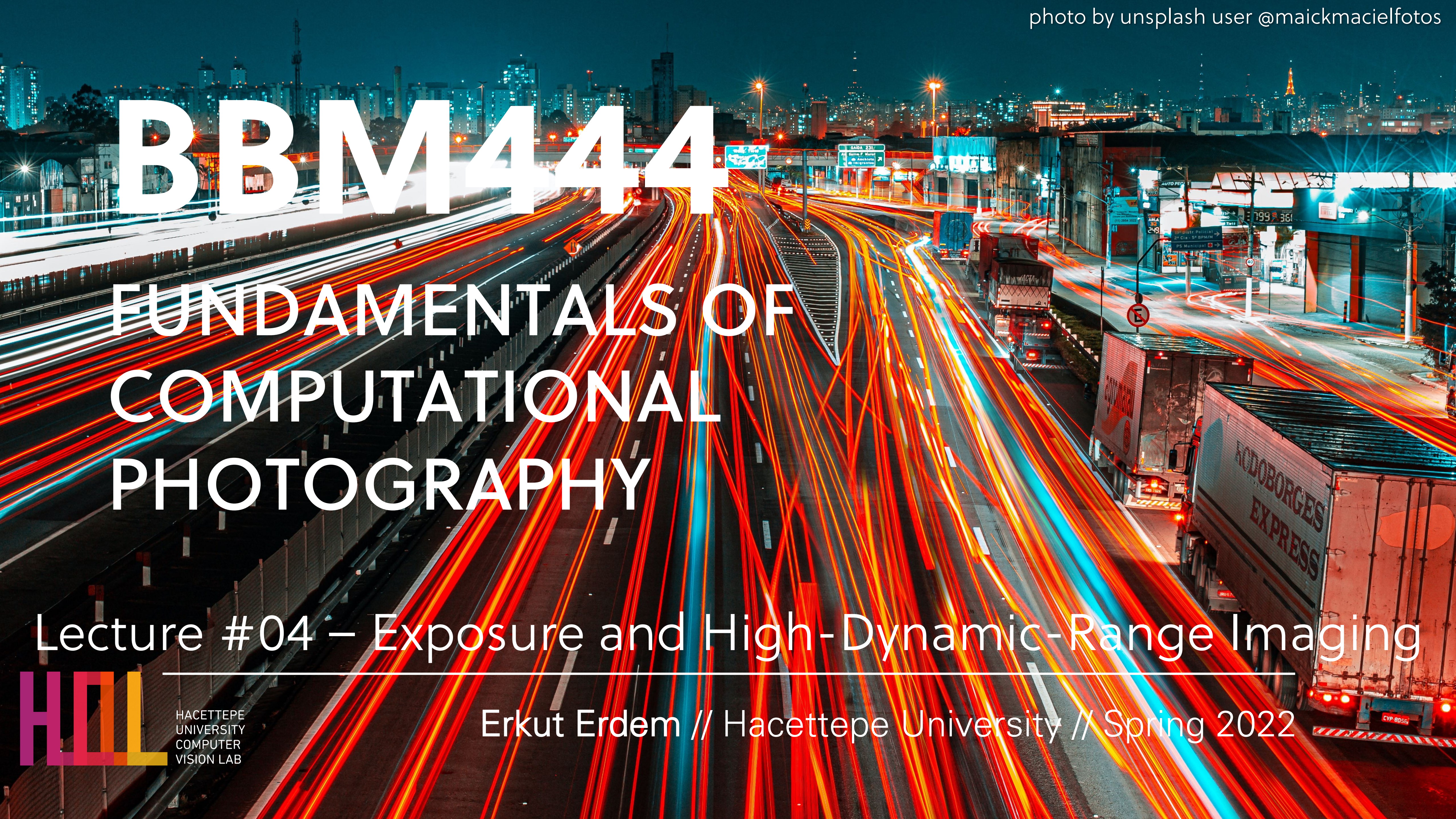


photo by unsplash user @maickmacielfotos

BBM444

FUNDAMENTALS OF COMPUTATIONAL PHOTOGRAPHY

Lecture #04 – Exposure and High-Dynamic-Range Imaging

HACETTEPE UNIVERSITY COMPUTER VISION LAB

Erkut Erdem // Hacettepe University // Spring 2022

Today's Lecture

- Controlling exposure
- High-dynamic-range imaging
- Tonemapping

Disclaimer: The material and slides for this lecture were borrowed from

—Ioannis Gkioulekas' 15-463/15-663/15-862 "Computational Photography" class

—Wojciech Jarosz's CS 89.15/189.5 "Computational Aspects of Digital Photography" class

—Derek Hoiem's CS 498 "Computational Photography" class

Light, exposure and dynamic range

- Exposure: how bright is the scene overall?
- Dynamic range: contrast in the scene
    - ratio of brightest to darkest intensity

Today's Lecture

- Controlling exposure
- High-dynamic-range imaging
- Tonemapping

Exposure control

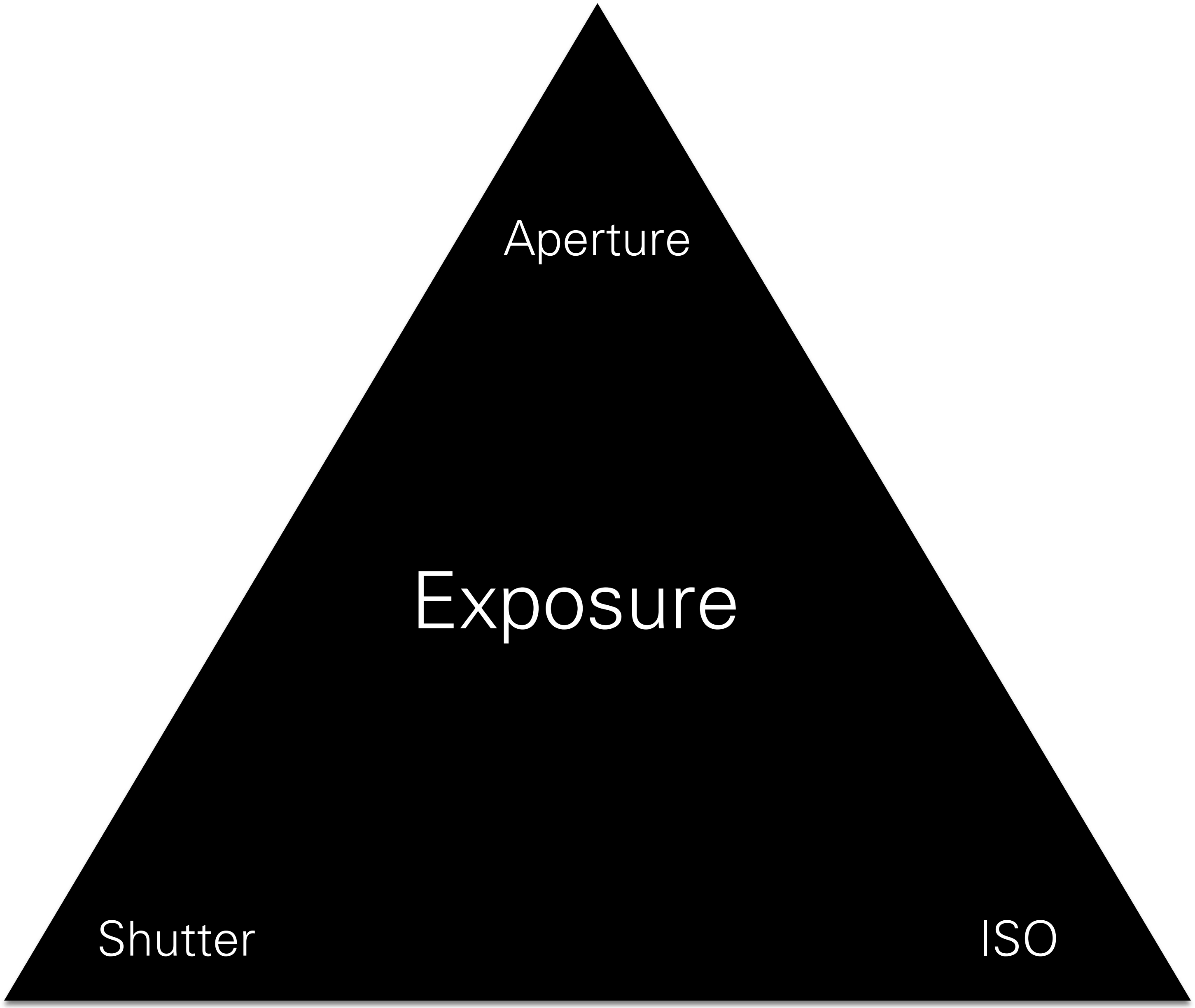
What is exposure?

Roughly speaking, the “brightness” of a captured image given a fixed scene.

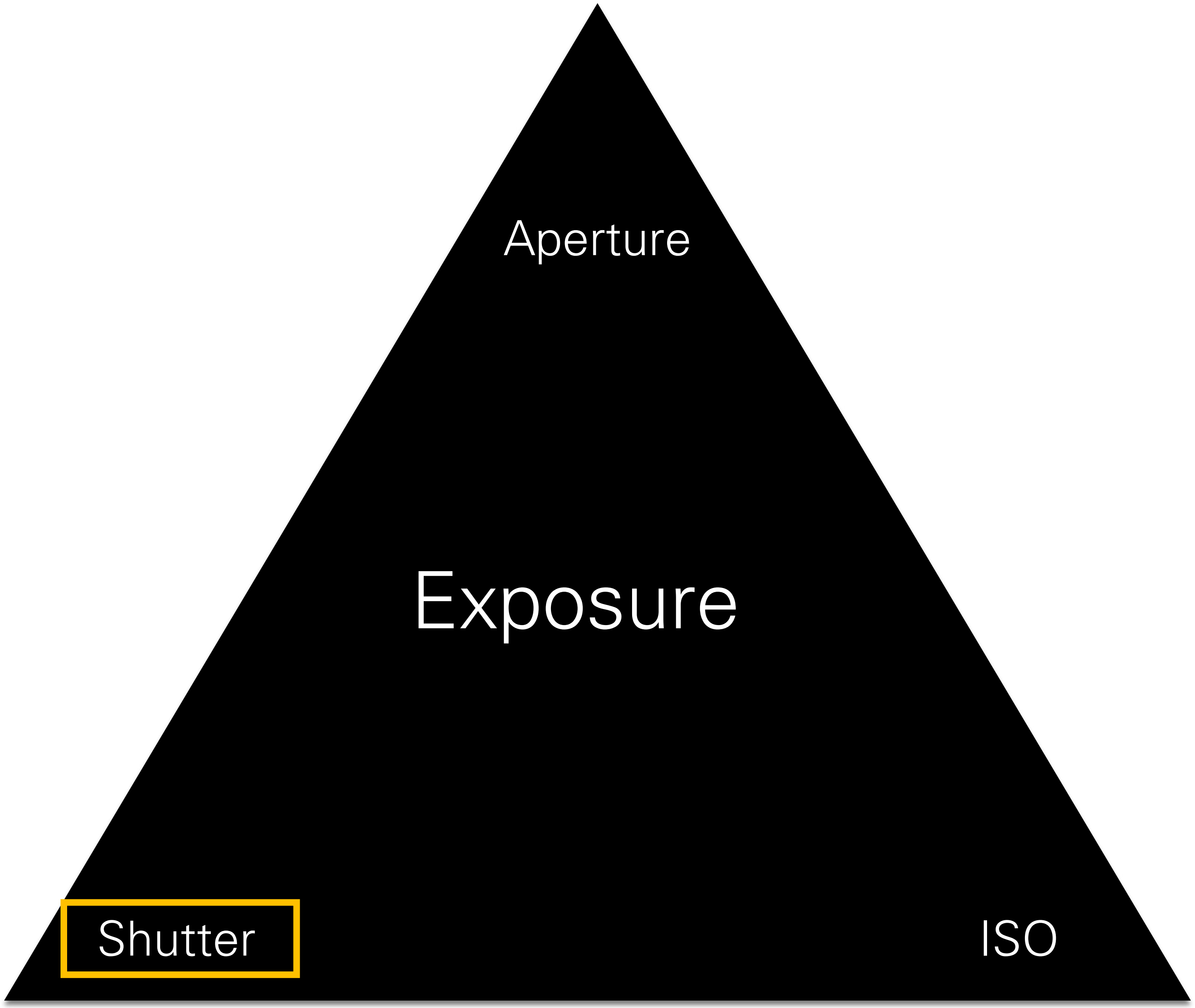
$$\text{Exposure} = \text{Gain} \times \text{Flux} \times \text{Time}$$

- Flux is controlled by the aperture.
- Time is controlled by the shutter speed.
- Gain is controlled by the ISO.

Exposure controls brightness of image

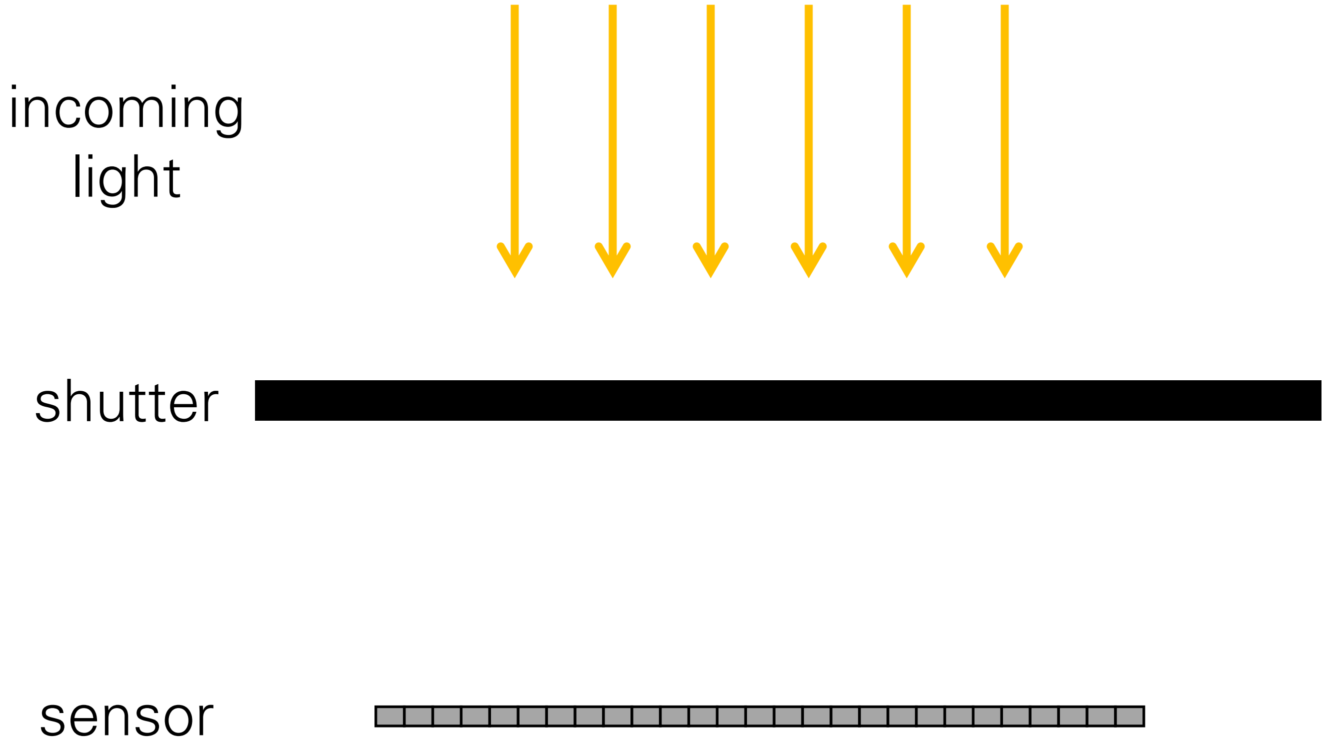


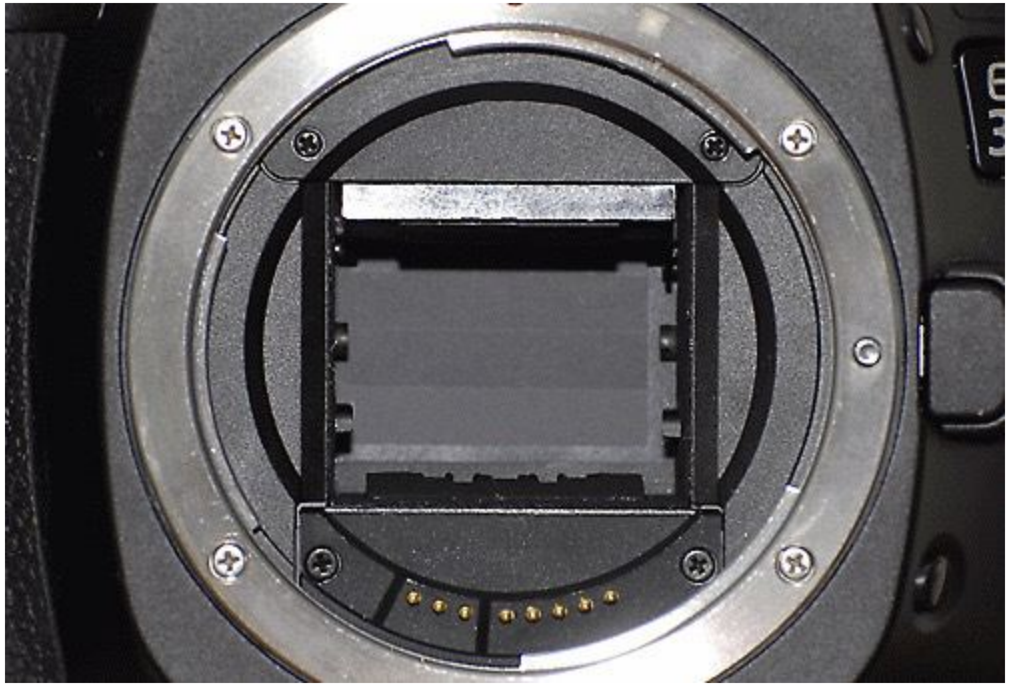
Exposure controls brightness of image



Shutter speed

Controls the **length of time** that shutter remains open.

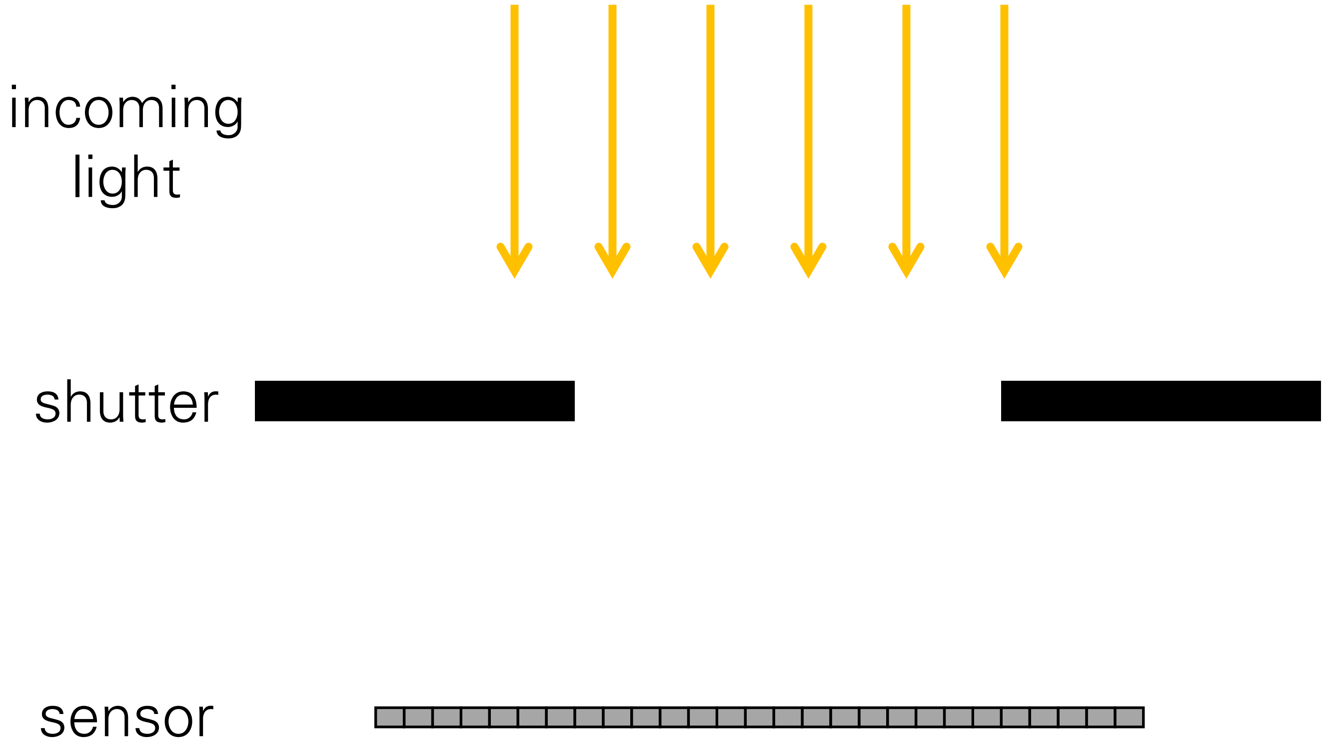


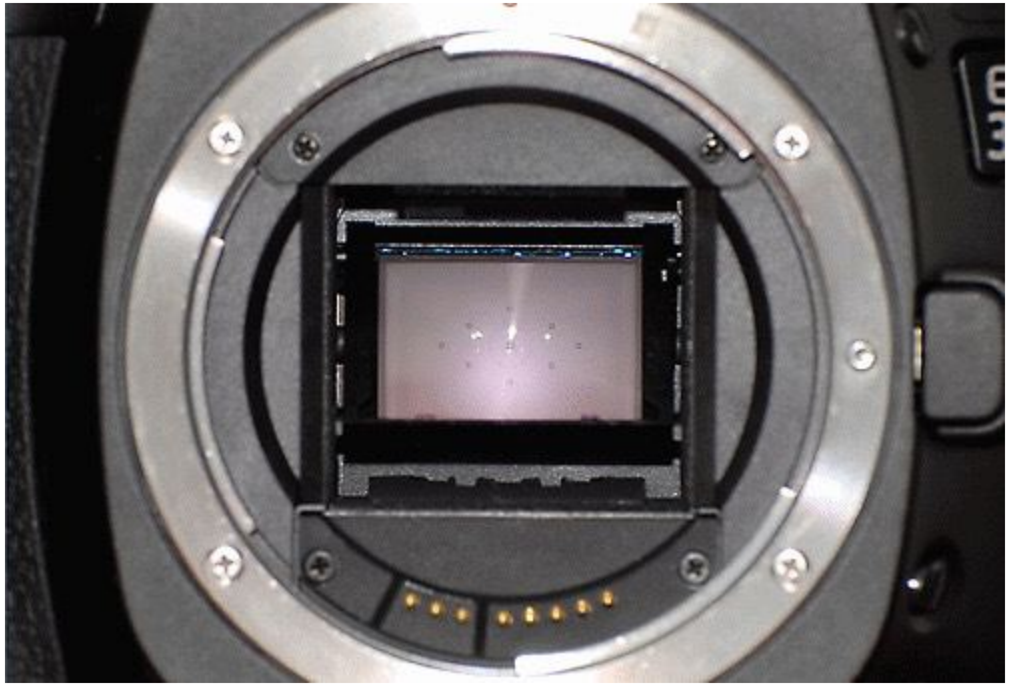


closed shutter

Shutter speed

Controls the **length of time** that shutter remains open.



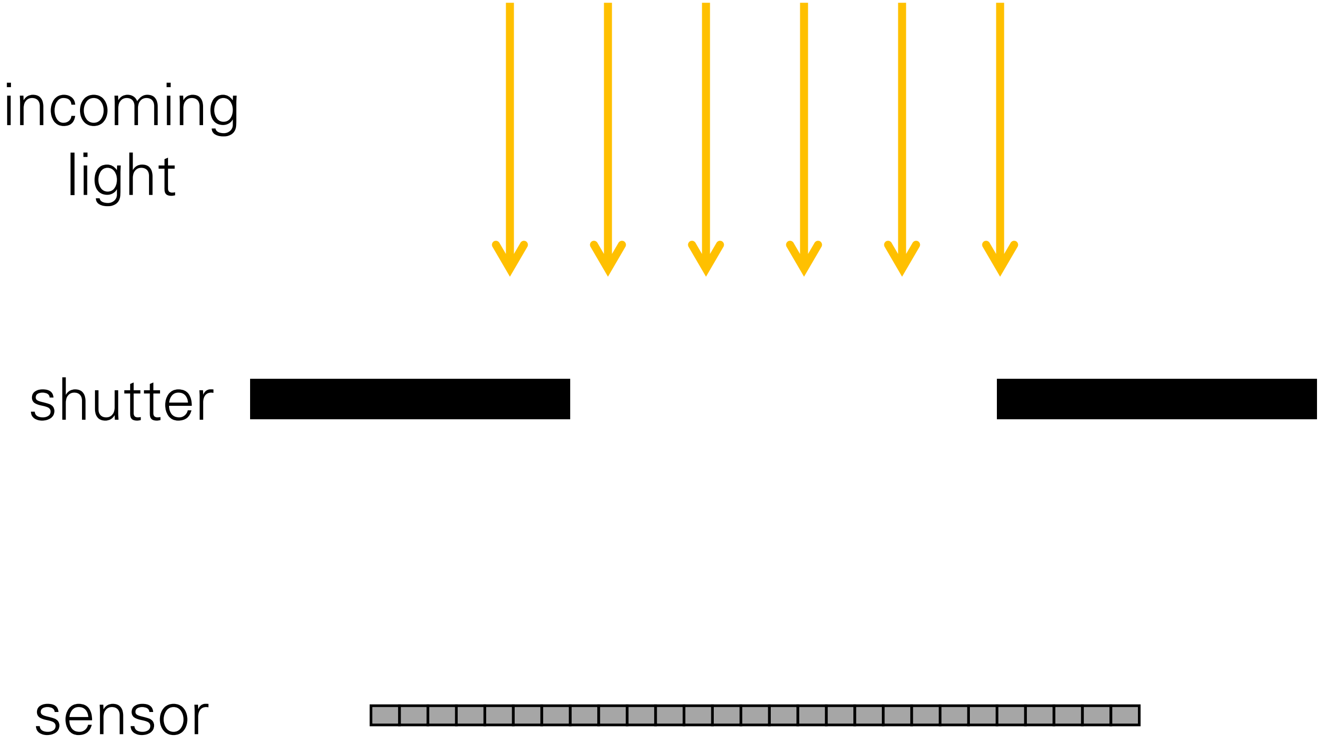


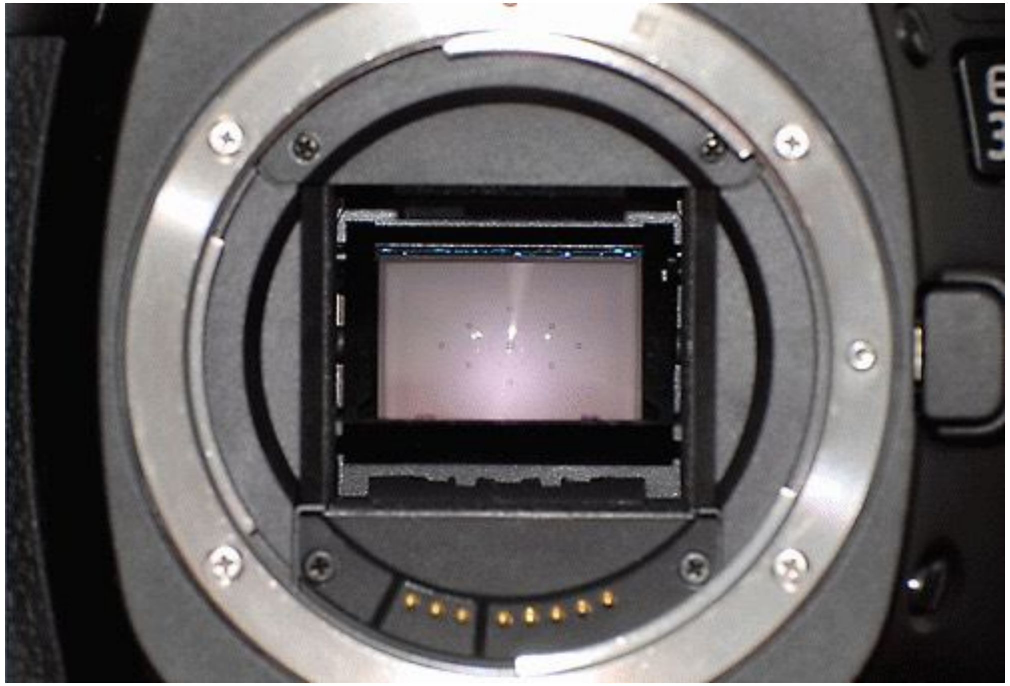
open shutter

Nikon D3s

Shutter speed

Controls the **period of time** that shutter remains open.



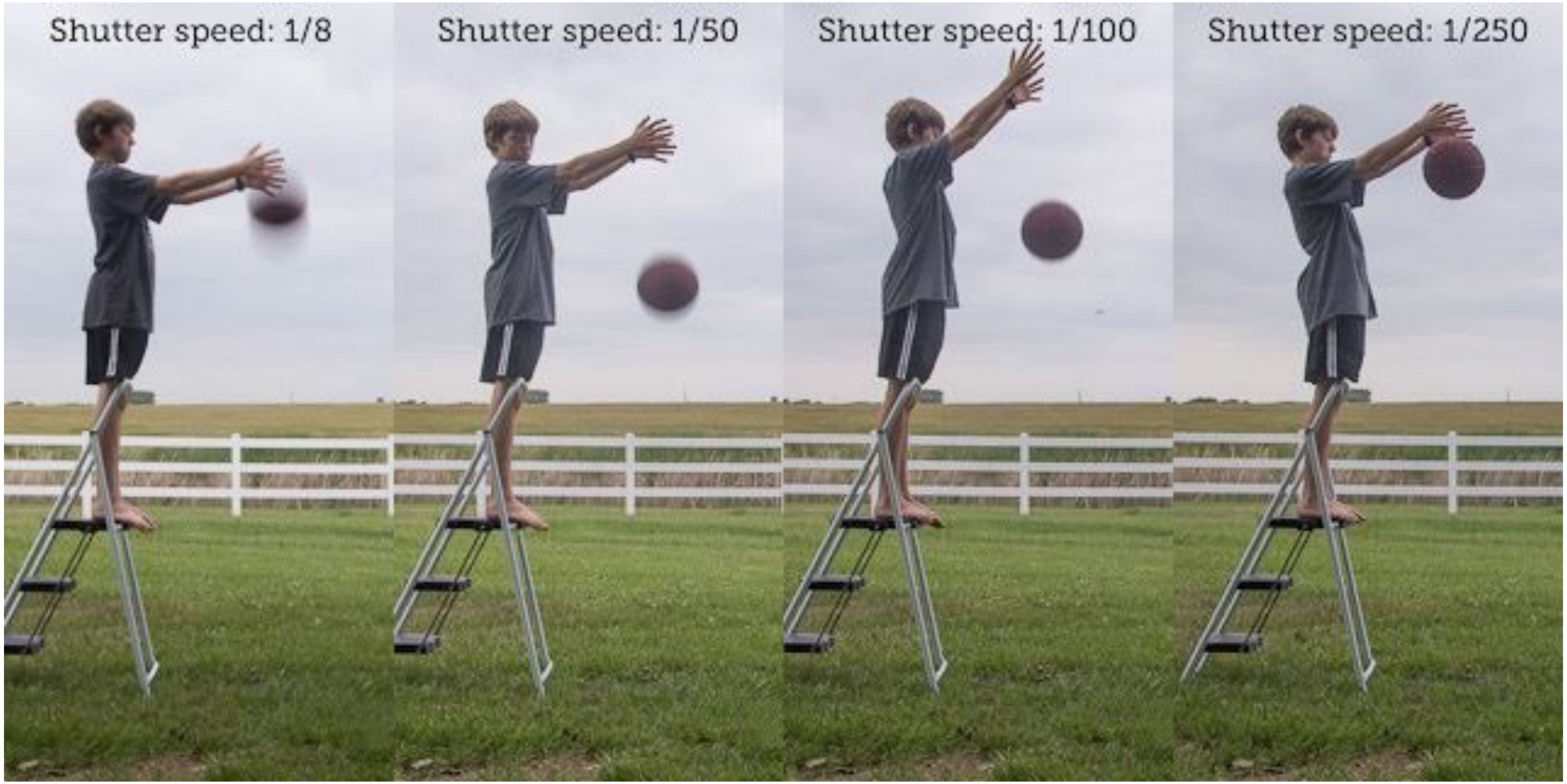


open shutter

What happens to the image as we increase shutter speed?

Side-effects of shutter speed

Moving scene elements appear blurry.

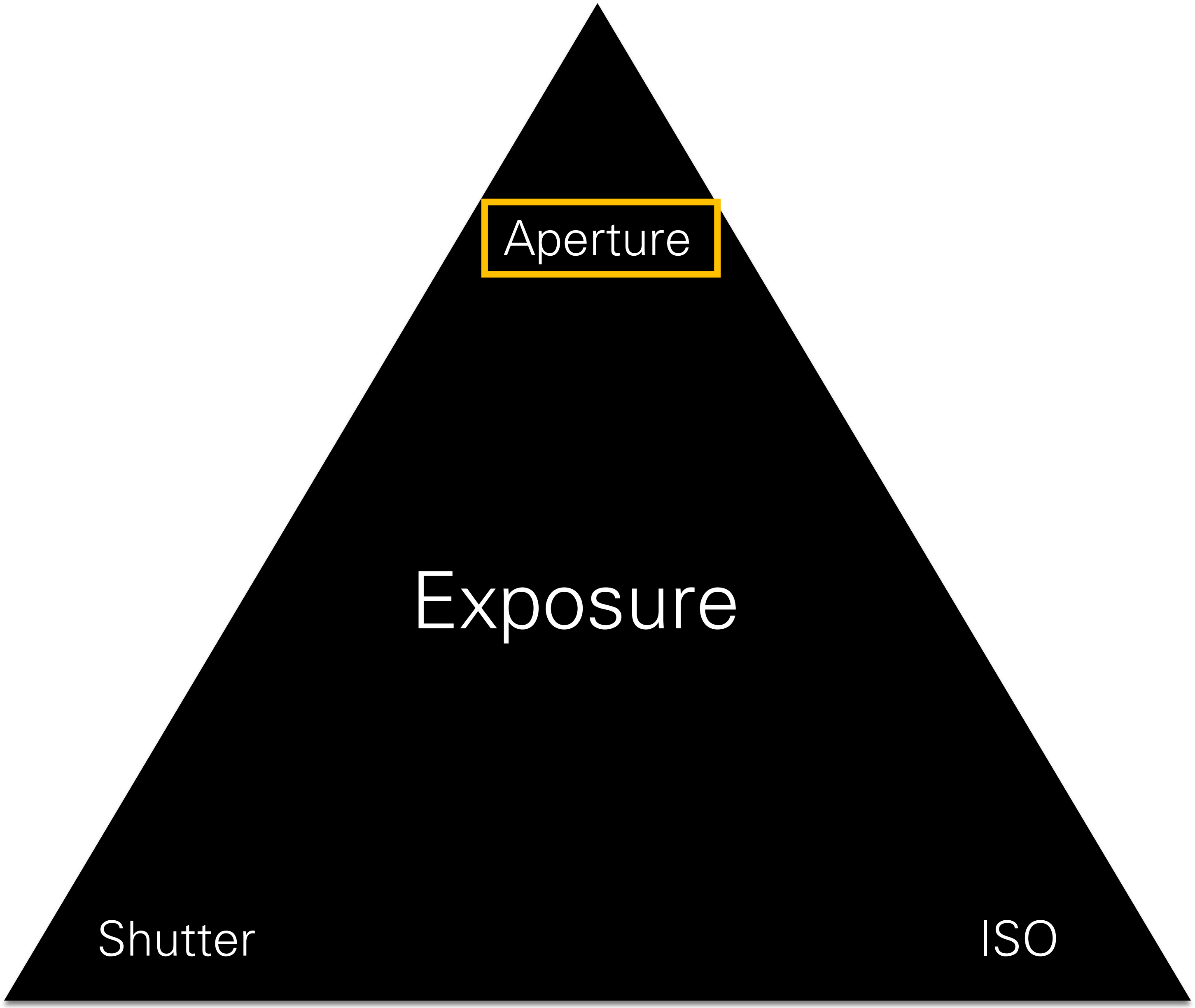


How can we "simulate" decreasing the shutter speed?

Motion deblurring

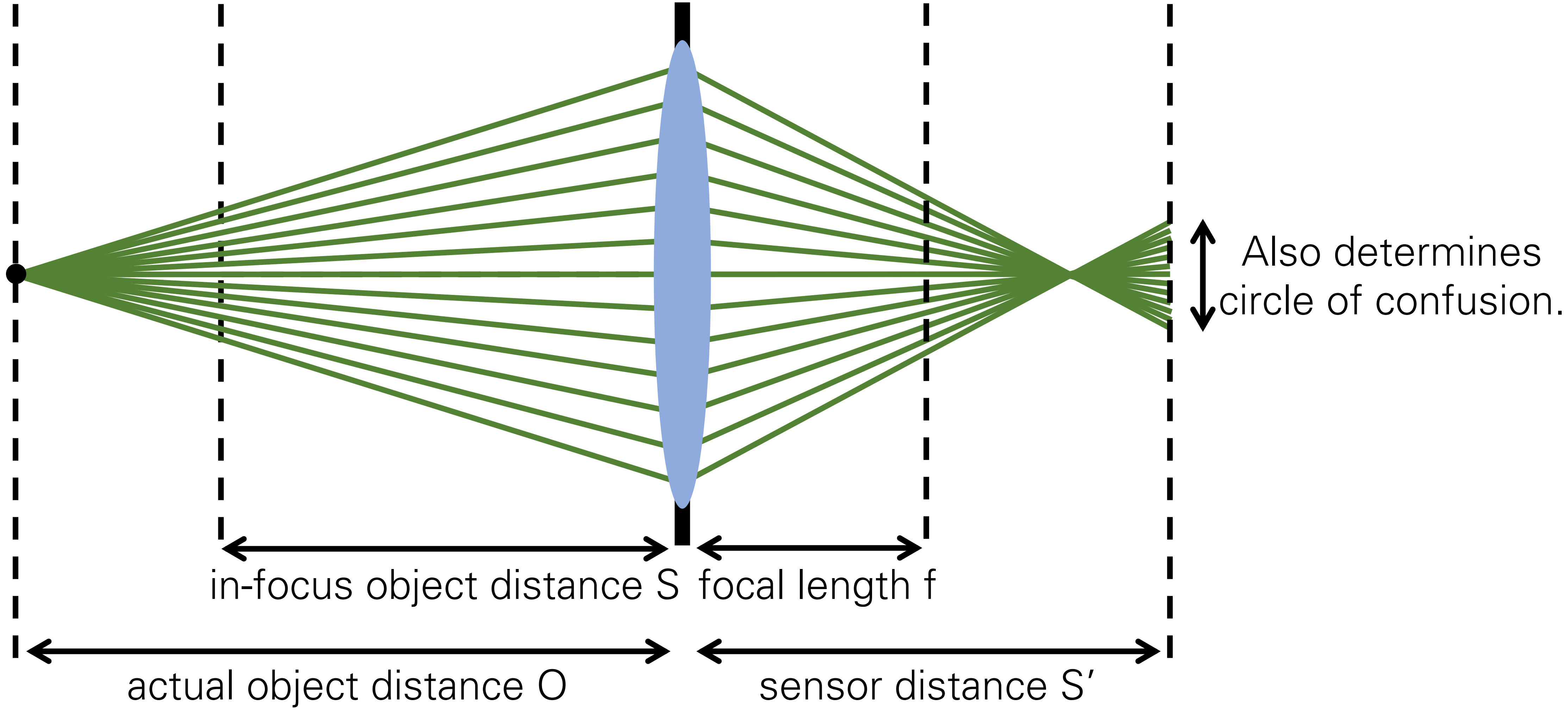
Shah et al. High-quality Motion Deblurring from a Single Image, SIGGRAPH 2008

Exposure controls brightness of image



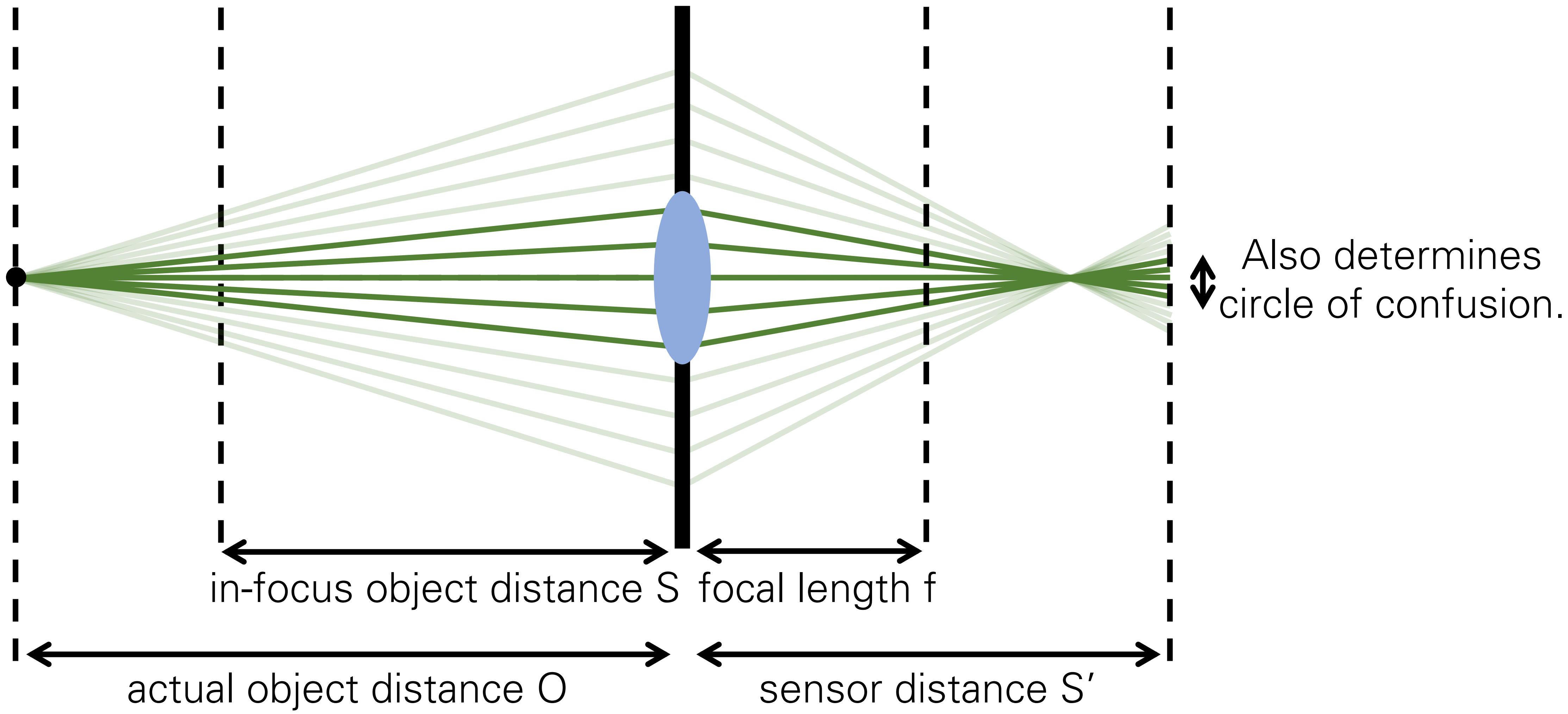
Aperture size

Controls area of lens that lets light pass through.



Aperture size

Controls area of lens that lets light pass through.



Aperture size

Most lenses have apertures of variable size.

- The size of the aperture is expressed as the “f-number”: The bigger this number, the smaller the aperture.

f / 1.4 f / 2.8 f / 4 f / 8 f / 16

You can see the aperture by removing the lens and looking inside it.

Side-effects of aperture size

Depth of field decreases as aperture size increases.

- Having a very sharp depth of field is known as “bokeh”.

How can we simulate bokeh?

How can we simulate bokeh?

Infer per-pixel depth, then blur with depth-dependent kernel.

- Example: Google camera "lens blur" feature

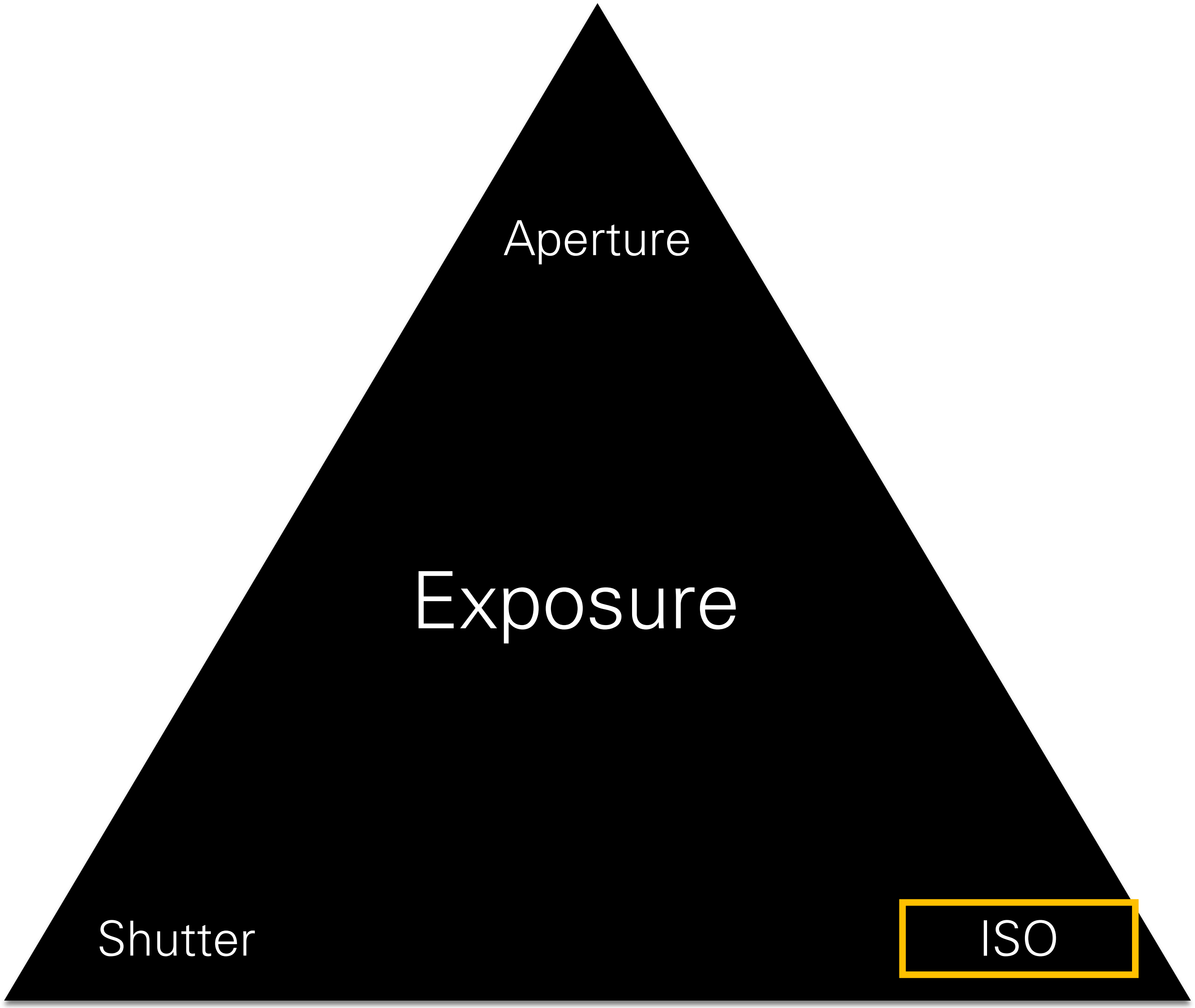
Barron et al., "Fast Bilateral-Space Stereo for Synthetic Defocus," CVPR 2015

How can we simulate bokeh?

Employ a learning-based strategy, i.e. an image-to-image translation model

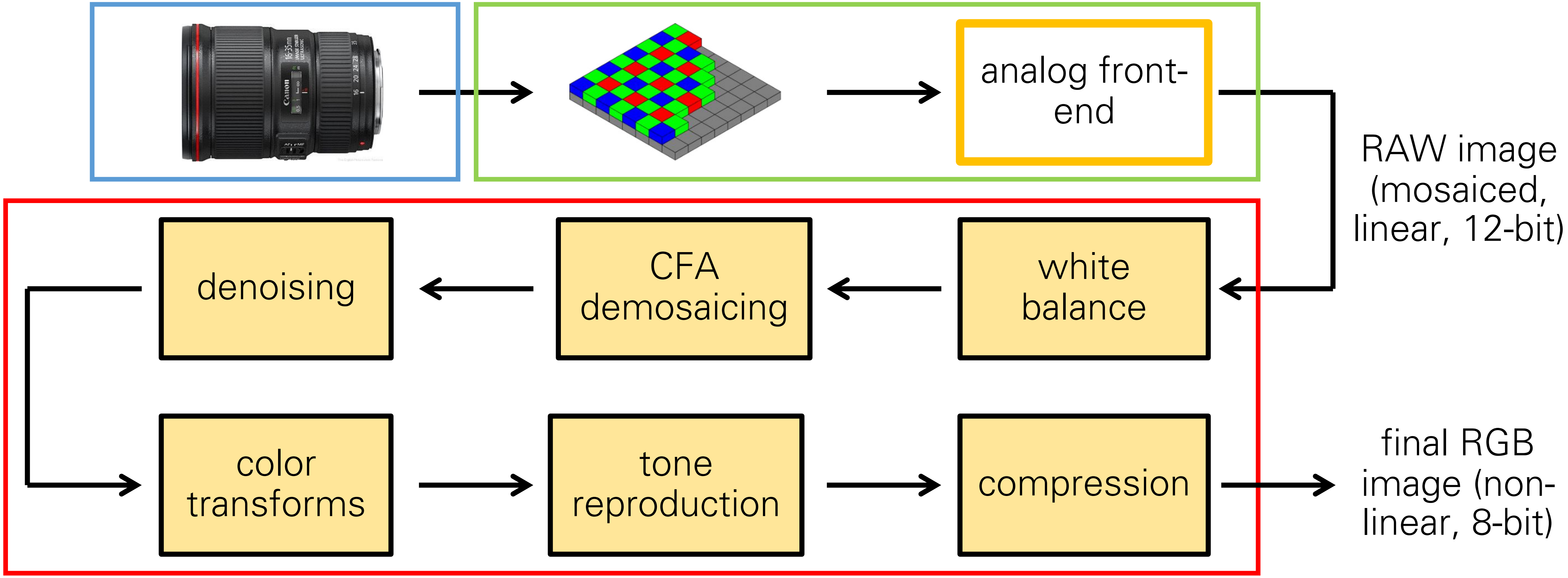
Zhang et al., “Synthetic Defocus and Look-Ahead Autofocus for Casual Videography,” SIGGRAPH 2019

Exposure controls brightness of image

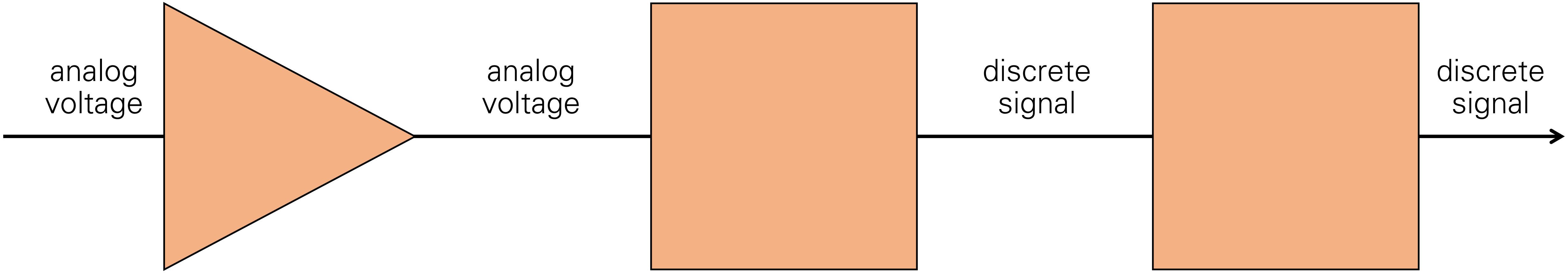


The (in-camera) image processing pipeline

The sequence of image processing operations applied by the camera's image signal processor (ISP) to convert a RAW image into a "conventional" image.



Analog front-end



analog amplifier (gain):

- gets voltage in range needed by A/D converter.
- accommodates ISO settings.
- accounts for vignetting.

analog-to-digital converter (ADC):

- depending on sensor, output has 10-16 bits.
- most often (?) 12 bits.

look-up table (LUT):

- corrects non-linearities in sensor's response function (within proper exposure).
- corrects defective pixels.

Side-effects of increasing ISO

Image becomes very grainy because noise is amplified.

ISO 80 ISO 800 ISO 1600

Note about the name ISO

ISO is not an acronym.

- It refers to the International Organization for Standardization.
- ISO comes from the Greek word ίσος, which means equal.
- It is pronounced (roughly) eye-zo, and should not be spelled out.

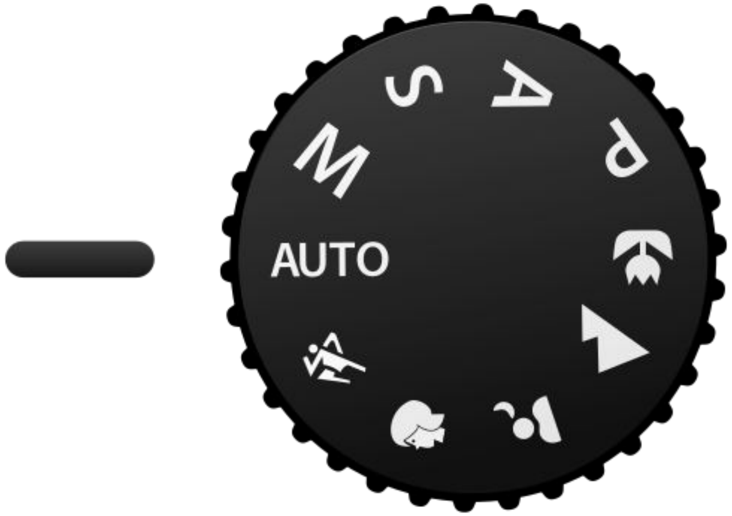
Camera modes

Aperture priority (“A”): you set aperture, camera sets everything else.
- Pros: Direct depth of field control.
- Cons: Can require impossible shutter speed (e.g. with f/1.4 for a bright scene).

Shutter speed priority (“S”): you set shutter speed, camera sets everything else.
- Pros: Direct motion blur control.
- Cons: Can require impossible aperture (e.g. when requesting a 1/1000 speed for a dark scene)

Automatic (“AUTO”): camera sets everything.
- Pros: Very fast, requires no experience.
- Cons: No control.

Manual (“M”): you set everything.
- Pros: Full control.
- Cons: Very slow, requires a lot of experience.



generic camera mode dial

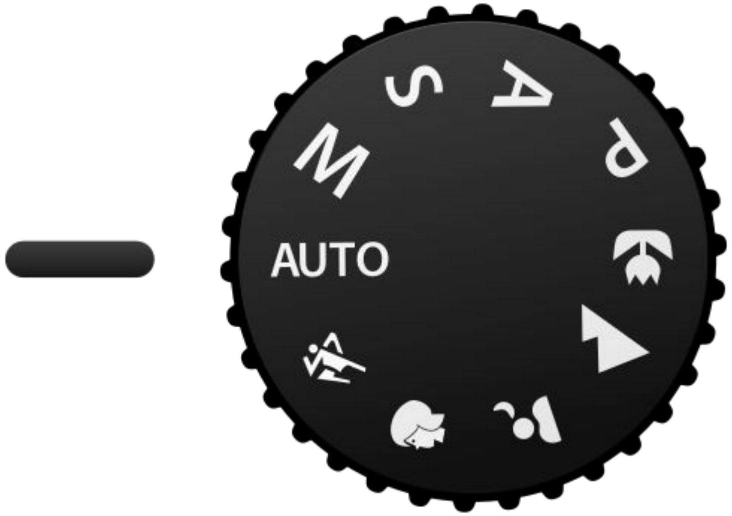
Camera modes

Aperture priority (“A”): you set aperture, camera sets everything else.
- Pros: Direct depth of field control.
- Cons: Can require impossible shutter speed (e.g. with f/1.4 for a bright scene).

Shutter speed priority (“S”): you set shutter speed, camera sets everything else.
- Pros: Direct motion blur control.
- Cons: Can require impossible aperture (e.g. when requesting a 1/1000 speed for a dark scene)

Automatic (“AUTO”): camera sets everything.
- Pros: Very fast, requires no experience.
- Cons: No control.

Manual (“M”): you set everything.
- Pros: Full control.
- Cons: Very slow, requires a lot of experience.

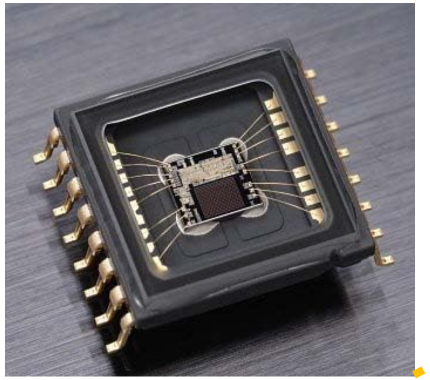


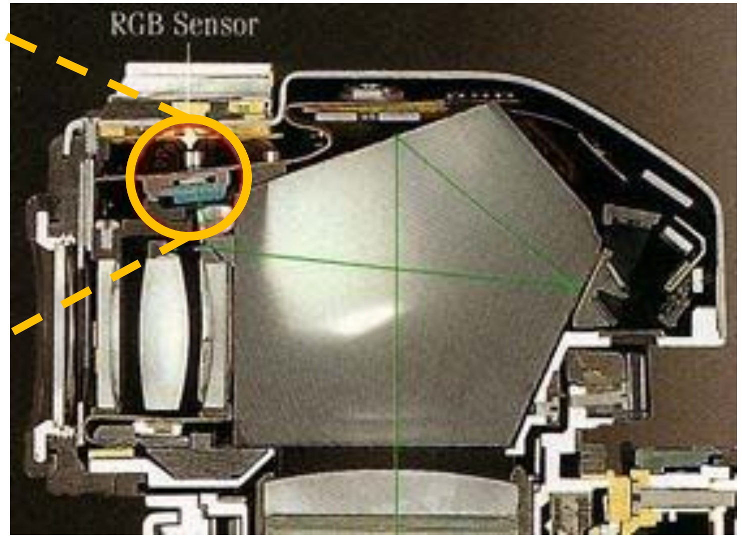
generic camera mode dial

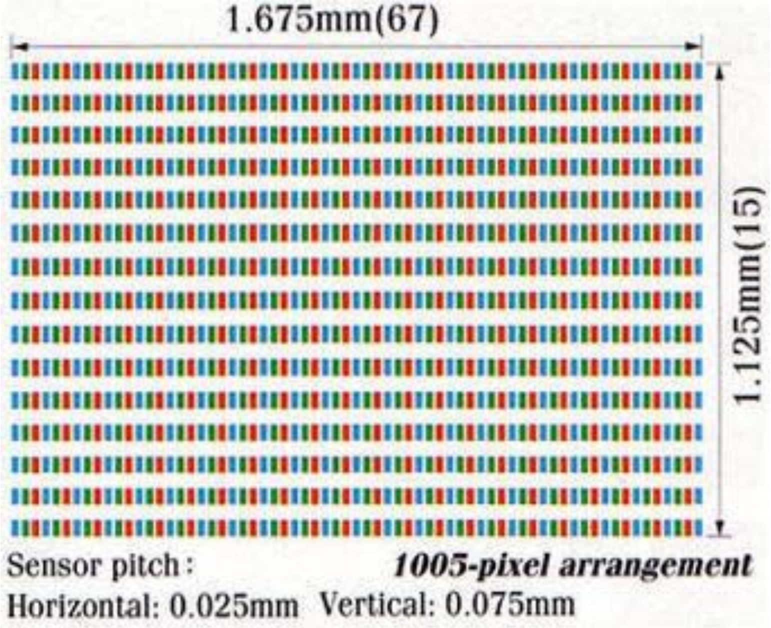
Light metering

Light metering in modern cameras

- SLR cameras use a separate low-resolution sensor that is placed at the focusing screen.







- Mirrorless cameras use measurements directly from the main sensor.

Light metering in modern cameras

- Measurements are averaged to produce a single intensity estimate, which is assumed to correspond to a scene of 18% reflectance (the "key").

- Exposure is set so that this average is exposed at the middle of the sensor's dynamic range.

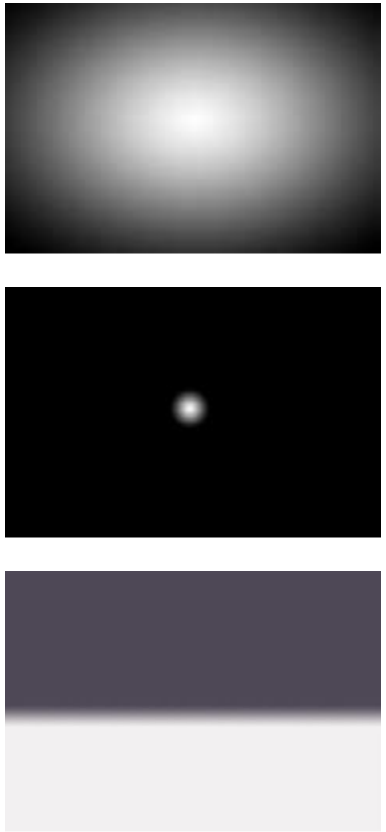
- Averaging can be done in many ways:

  1. Center-weighed.

  2. Spot.

  3. Scene-specific preset (portrait, landscape, horizon).

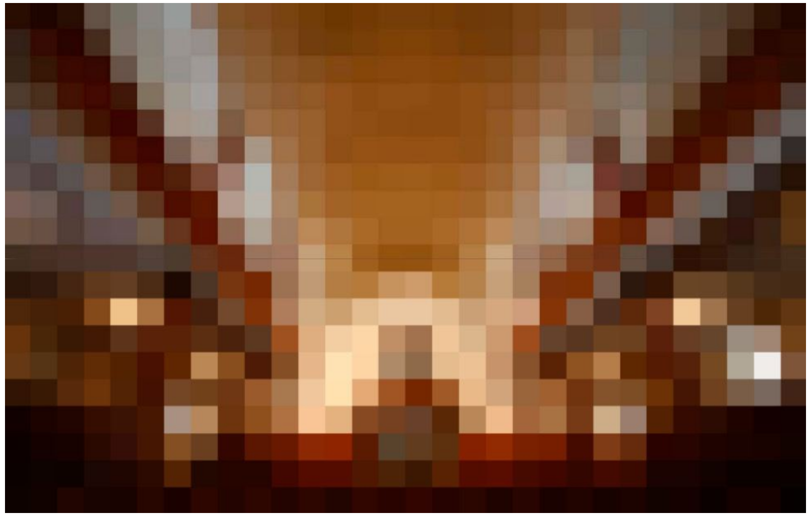
  4. "Intelligently" using proprietary algorithm.

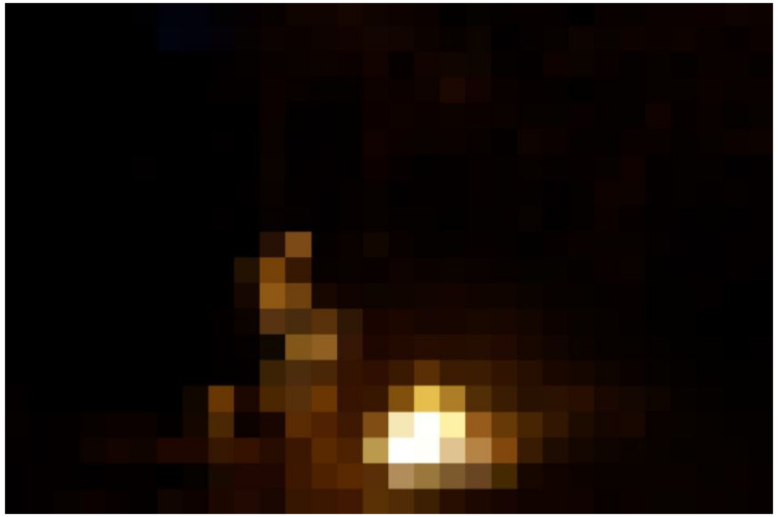


Metering challenges: low resolution

Low-resolution can make it difficult to correctly meter the scene and set exposure.

- In which of these scenes is it OK to let the brightest pixels be overexposed?

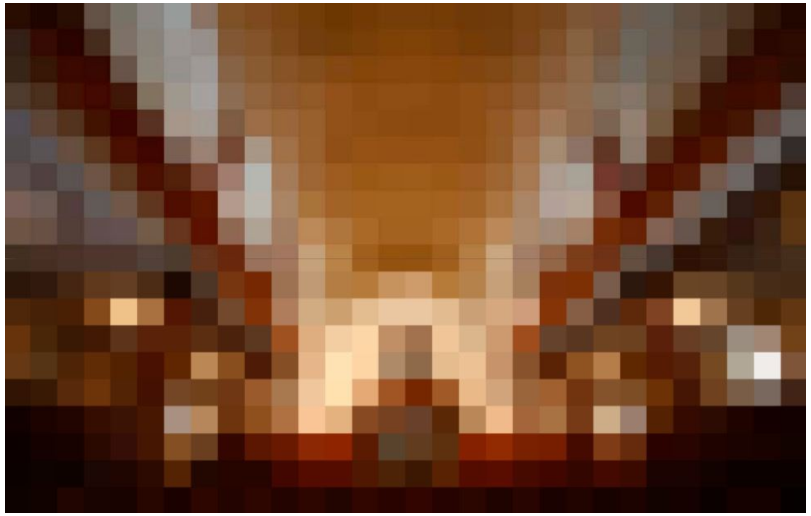
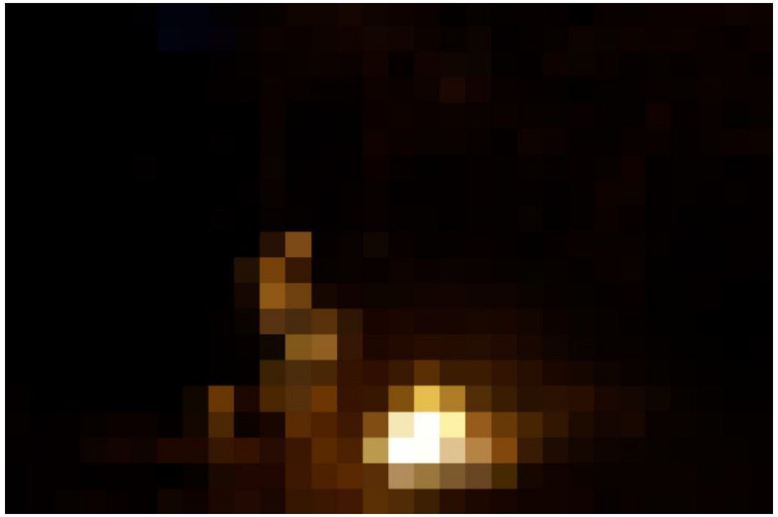




Metering challenges: low resolution

Low-resolution can make it difficult to correctly meter the scene and set exposure.

- In which of these scenes is it OK to let the brightest pixels be overexposed?



Today’s Lecture

- Controlling exposure
- High-dynamic-range imaging
- Tonemapping

Light, exposure and dynamic range

- Exposure: how bright is the scene overall?
- Dynamic range: contrast in the scene
    - ratio of brightest to darkest intensity

Our devices do not match the world

The world has a high dynamic range

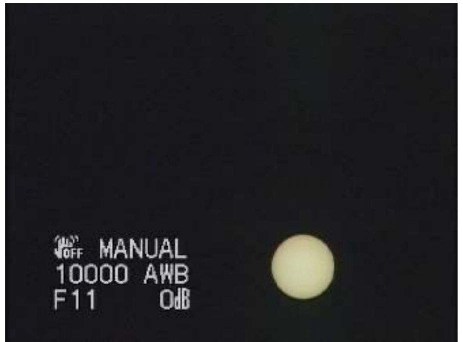
1

1500

25,000

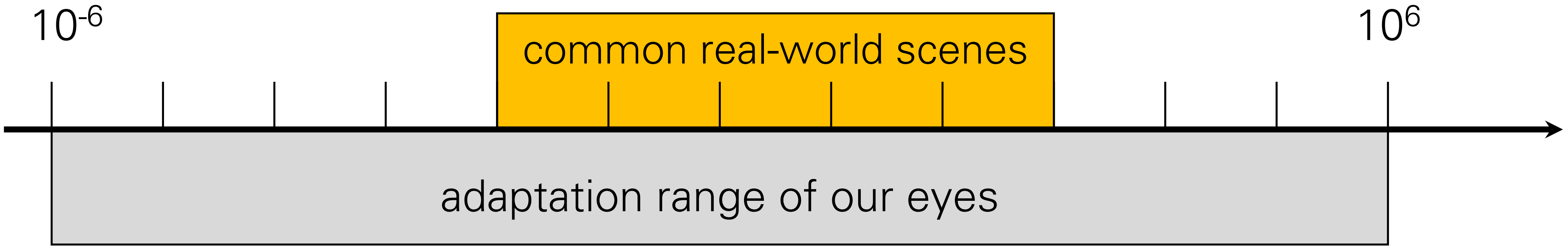
400,000

2,000,000,000

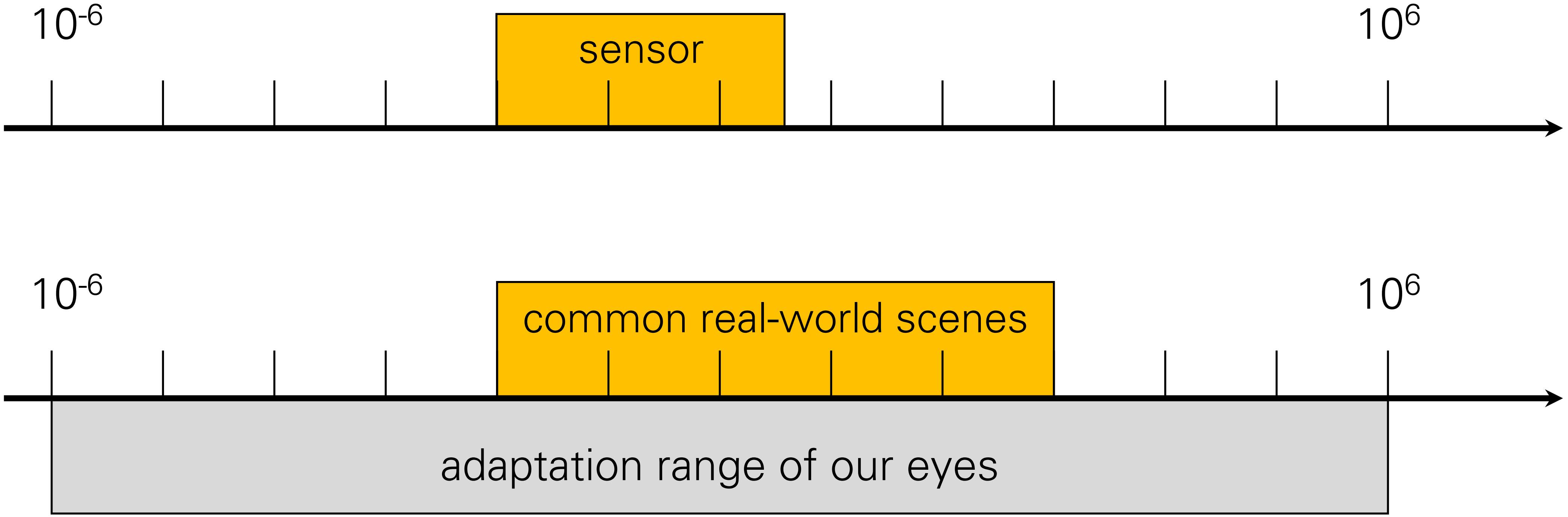


Relative brightness of different scenes, ranging from 1 inside a dark room lit by a monitor to 2,000,000 looking at the Sun.

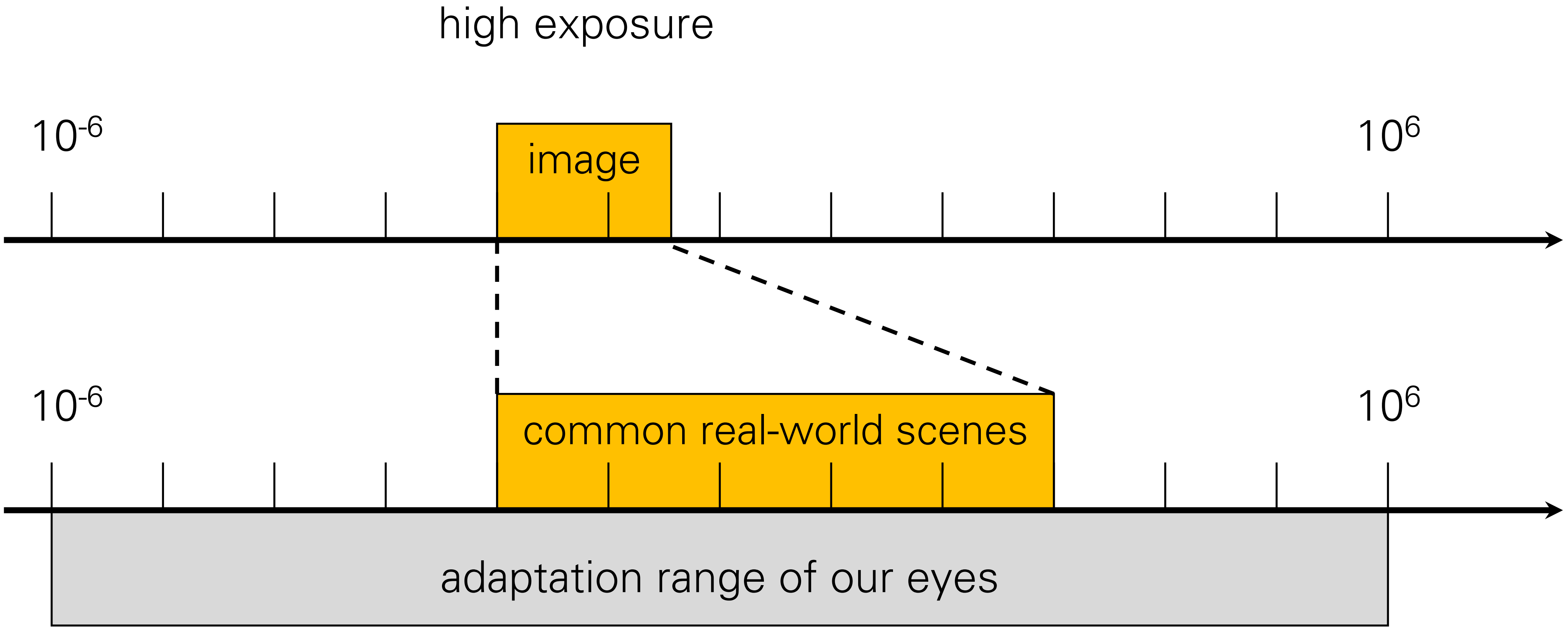
The world has a high dynamic range



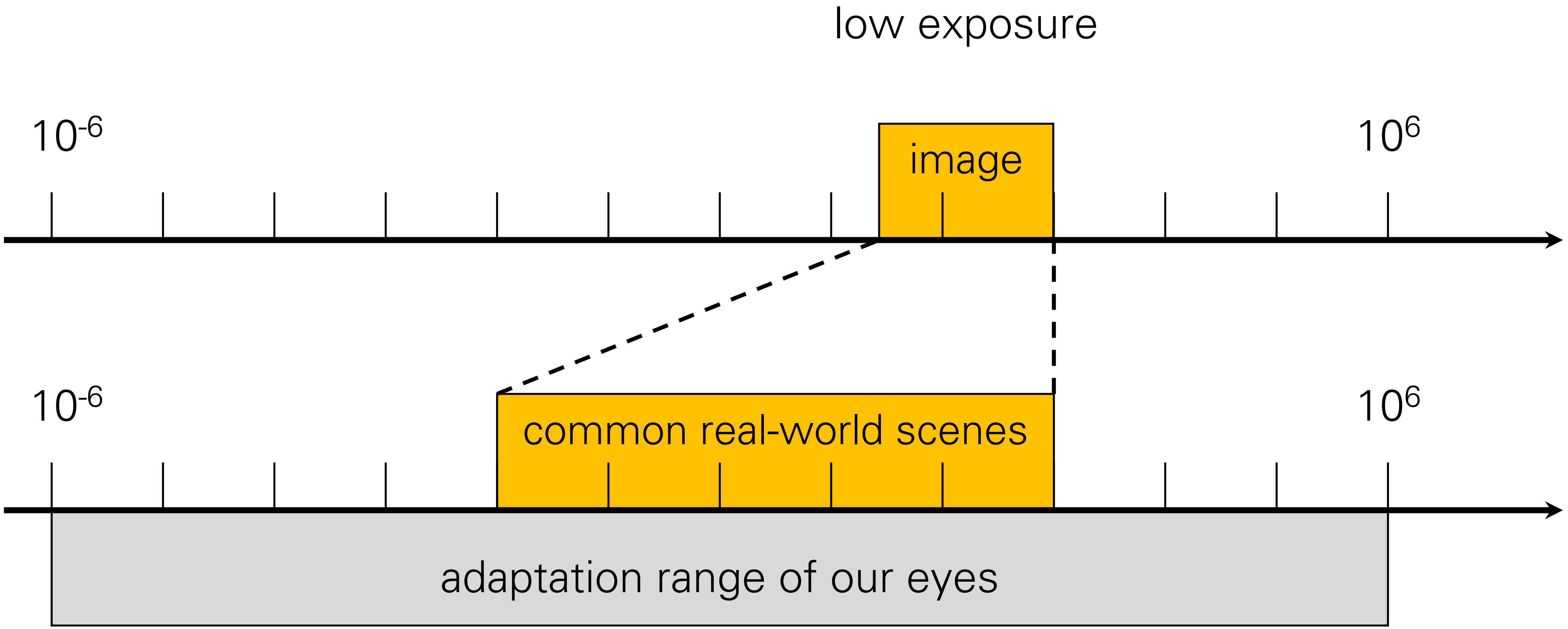
(Digital) sensors also have a low dynamic range



(Digital) images have an even lower dynamic range



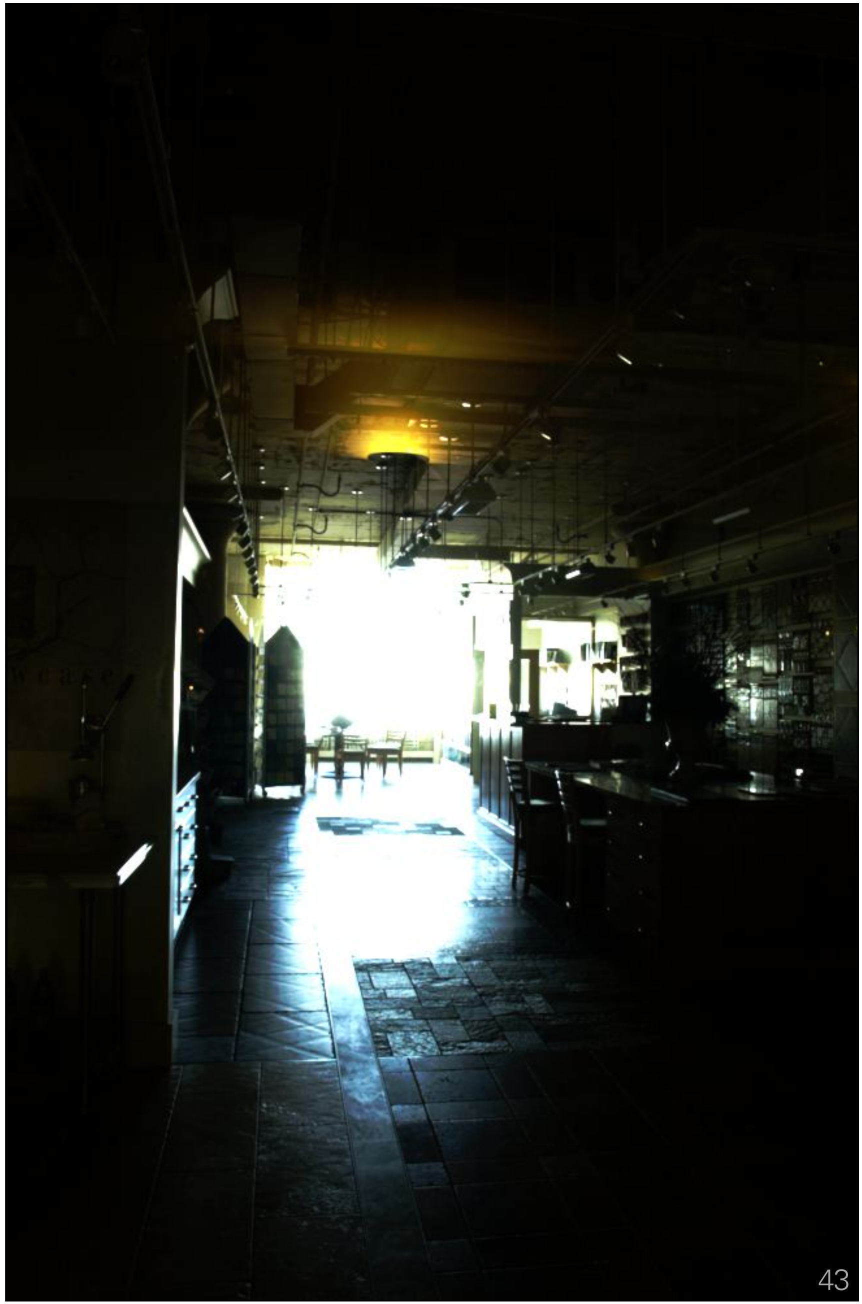
(Digital) images have an even lower dynamic range



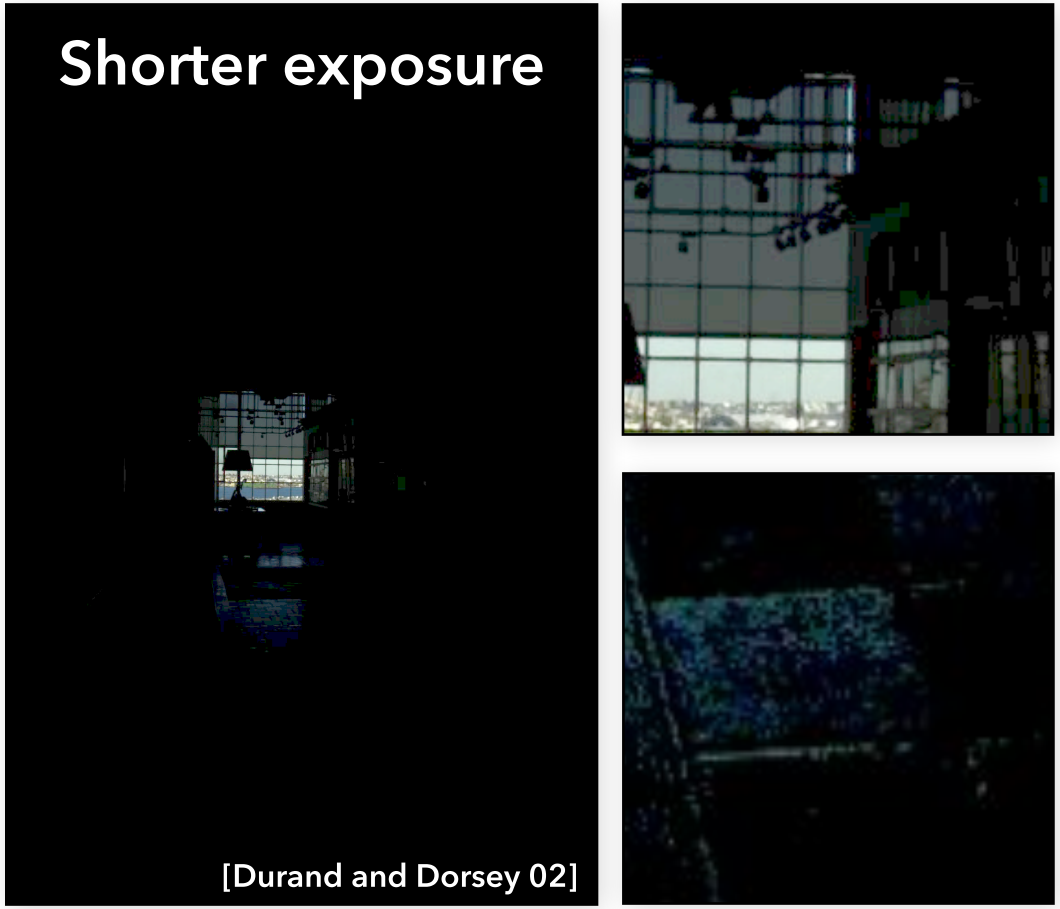
The dynamic range challenge

Sun overexposed
Foreground too dark

Inside is too dark
Outside is too bright



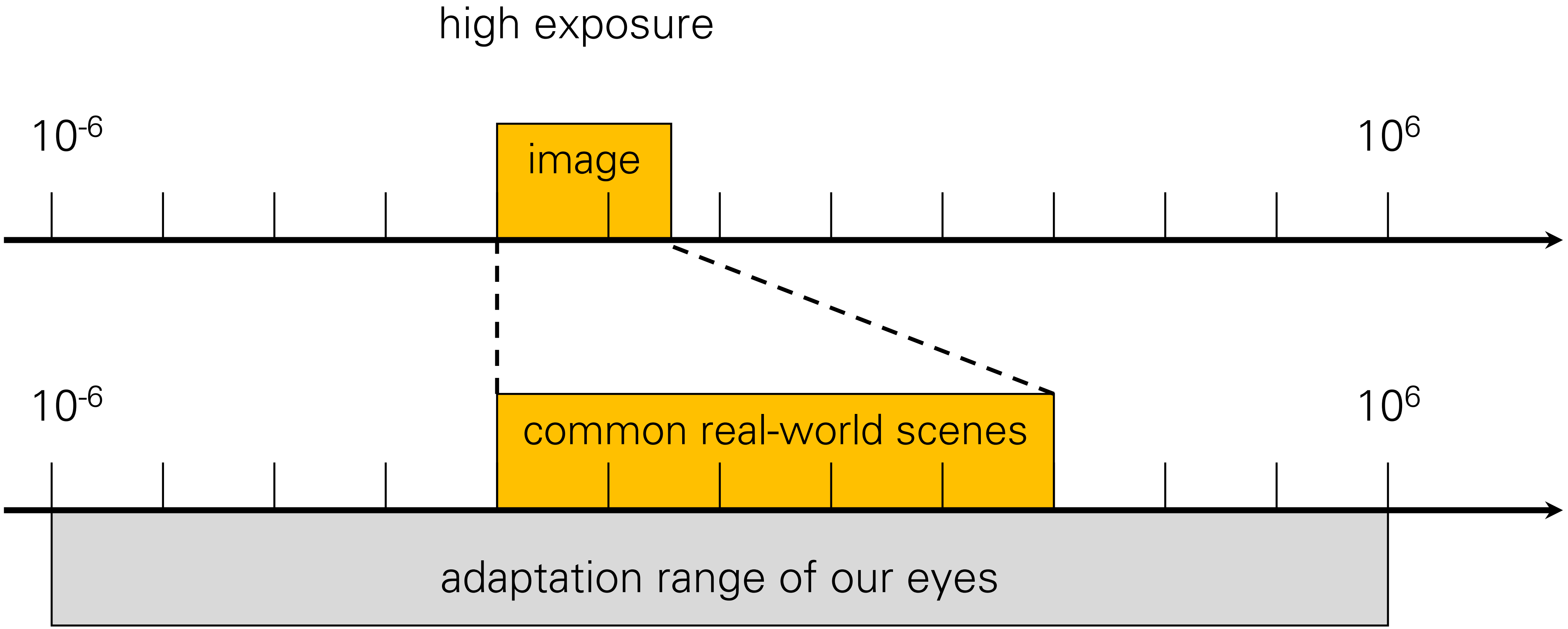
Low Dynamic Range (LDR)



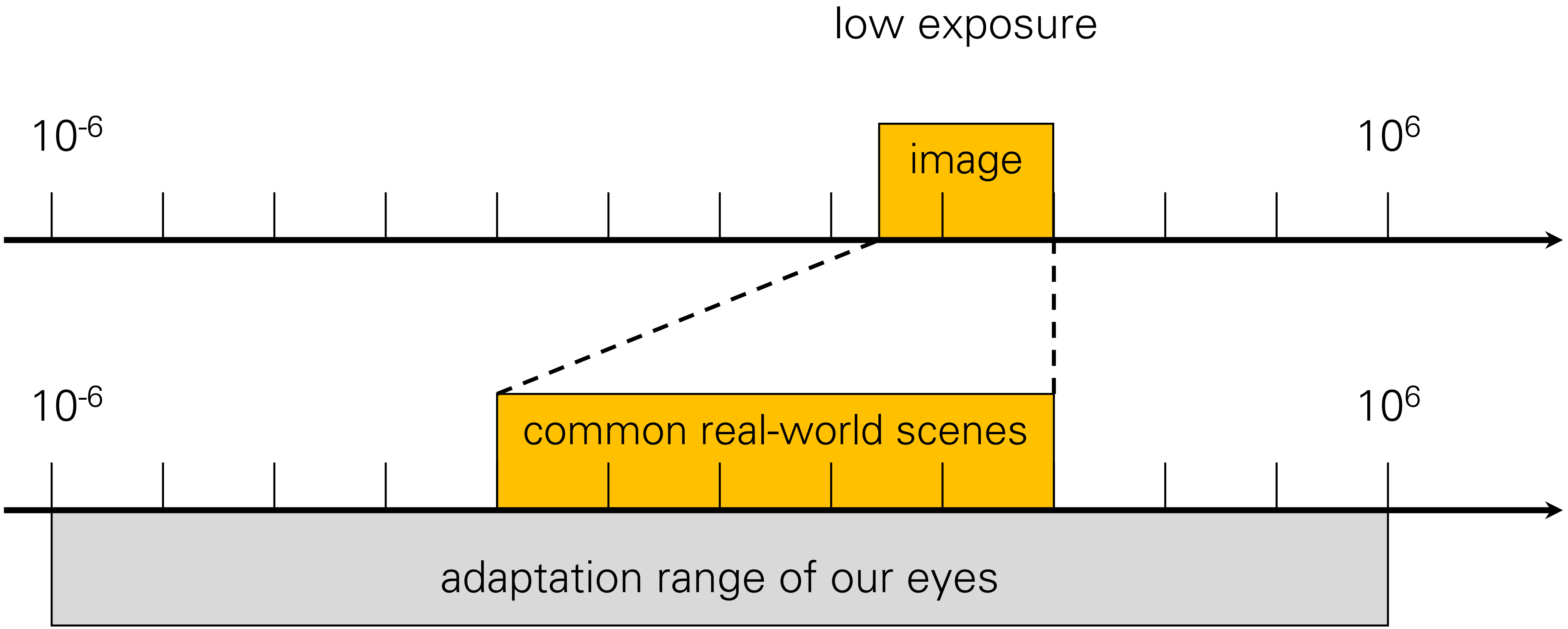
✓ detail in shadows
✗ clipped highlights

✓ detail in highlights
✗ noisy/clipped shadows

(Digital) images have an even lower dynamic range

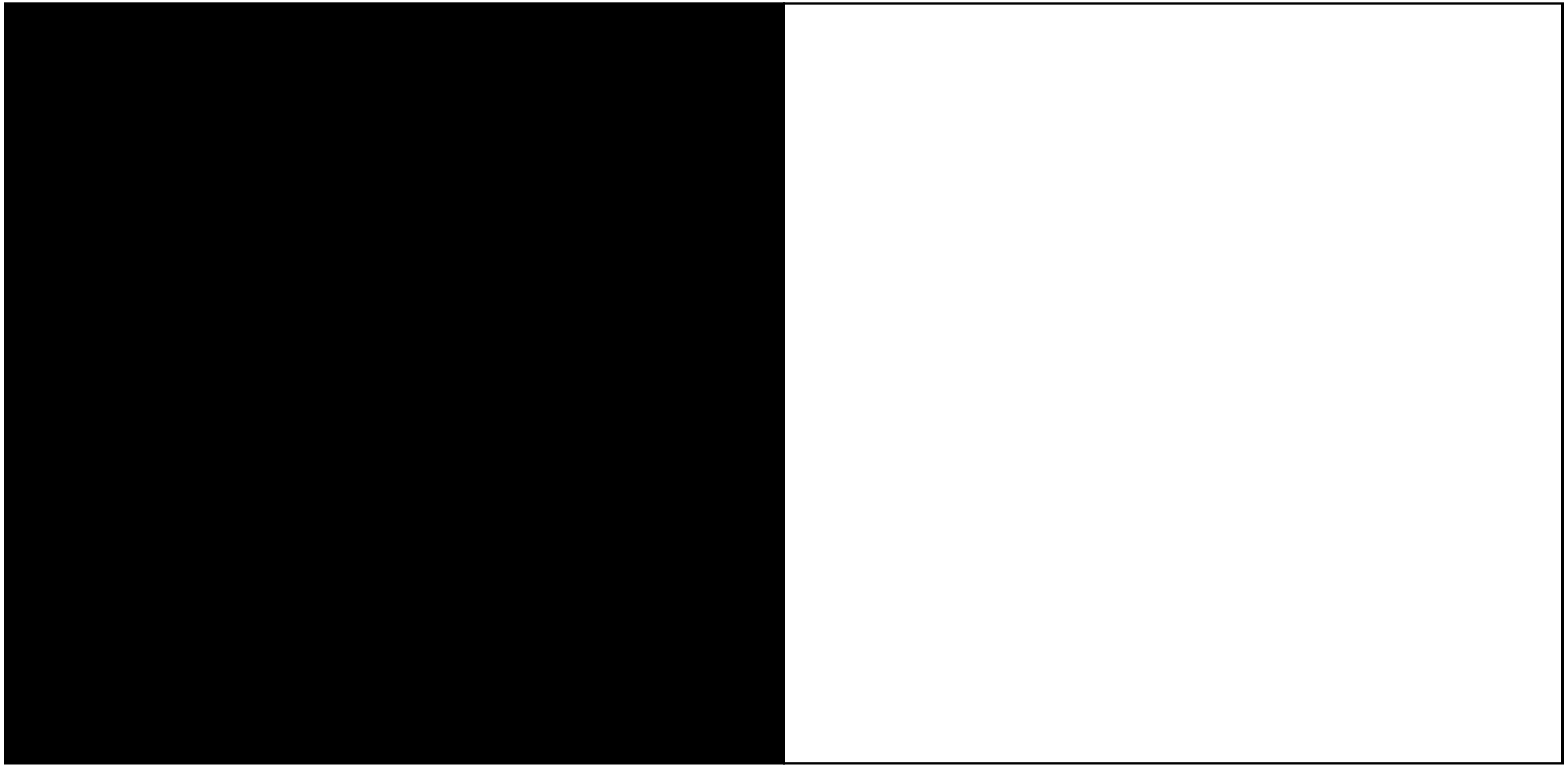


(Digital) images have an even lower dynamic range



(Digital) images have an even lower dynamic range

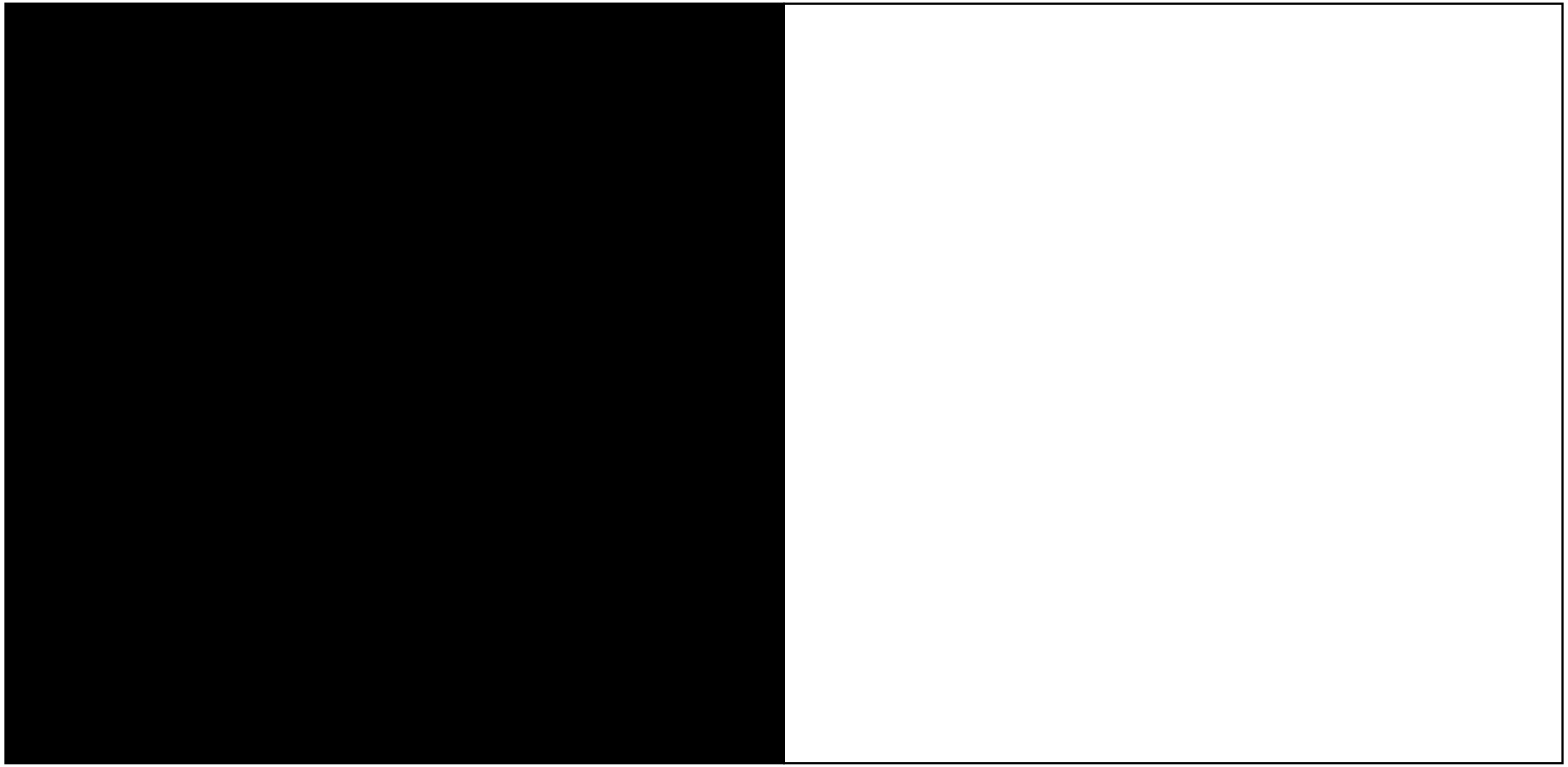
Any guesses about the dynamic range of a standard 0-255 image?



pure black pure white

(Digital) images have an even lower dynamic range

Any guesses about the dynamic range of a standard 0-255 image?



about 50x brighter

pure black

pure white

Our devices do not match the real world

• 10:1	photographic print (higher for glossy paper)
• 20:1	artist's paints
• 200:1	slide film
• 500:1	negative film
• 1000:1	LCD display
• 2000:1	digital SLR (at 12 bits)
• 100000:1	real world

Two challenges:

1. HDR imaging – which parts of the world do we measure in the 8-14 bits available to our sensor?
2. Tonemapping – which parts of the world do we show in the 4-10 bits available to our display?

Our devices do not match the real world

• 10:1	photographic print (higher for glossy paper)
• 20:1	artist's paints
• 200:1	slide film
• 500:1	negative film
• 1000:1	LCD display
• 2000:1	digital SLR (at 12 bits)
• 100000:1	real world

HDR imaging and tonemapping are distinct techniques with different goals

Two challenges:

HDR imaging compensates for sensor limitations

1. HDR imaging – which parts of the world do we measure in the 8-14 bits available to our sensor?
2. Tonemapping – which parts of the world do we show in the 4-10 bits available to our display?

Tonemapping compensates for display limitations

High dynamic range imaging


HYATT


HYATT


HYATT

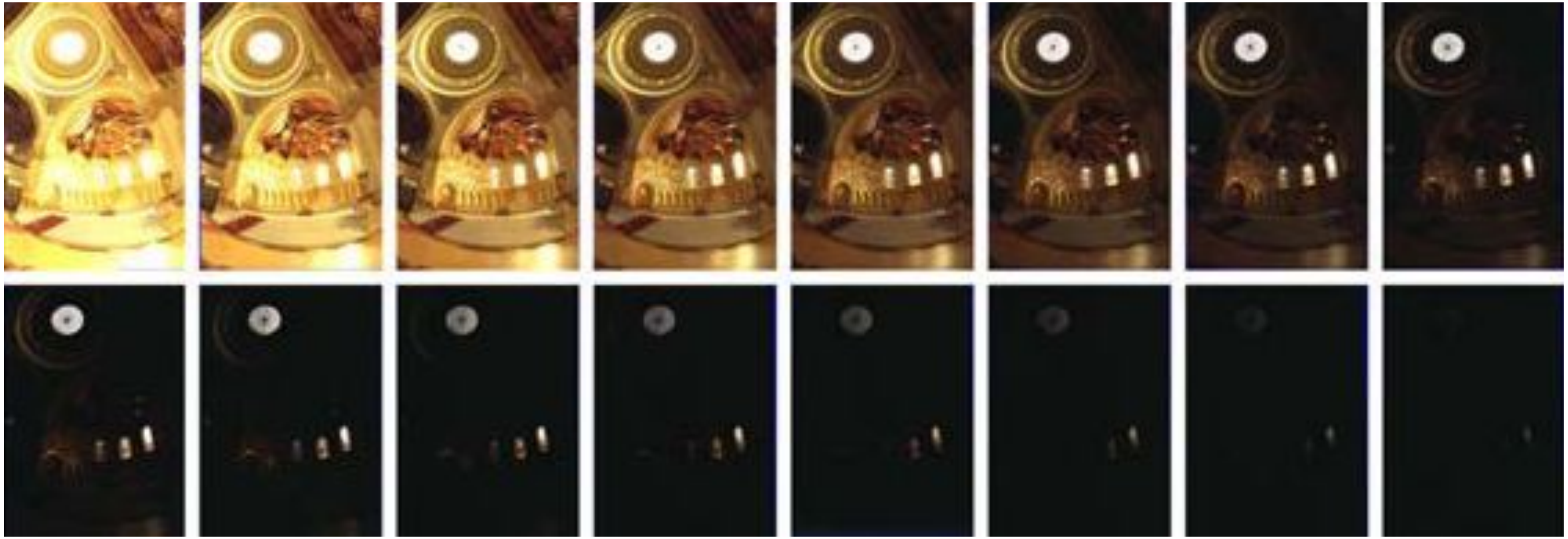

THE INTERNATIONAL FUR EXCHANGE

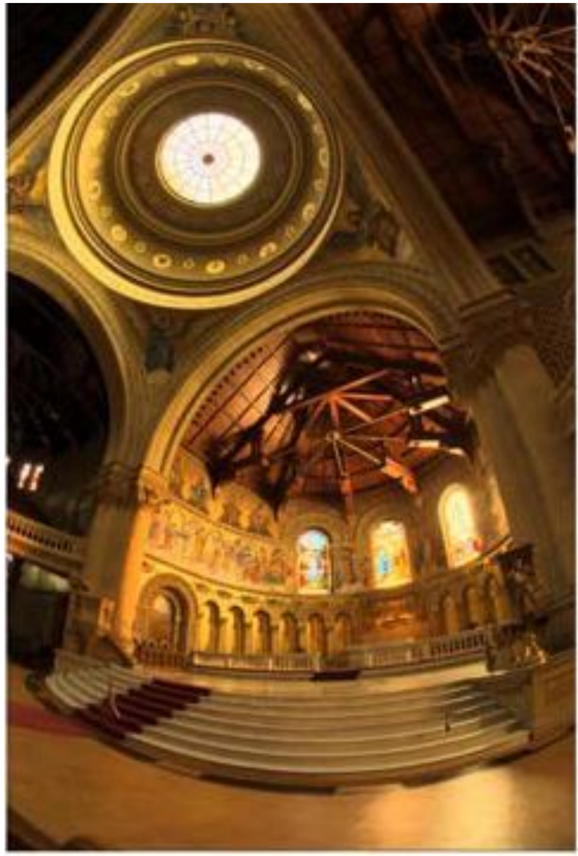

HYATT


HYATT

Key idea

1. Exposure bracketing: Capture multiple LDR images at different exposures





2. Merging: Combine them into a single HDR image

“Sunset from Rigi Kaltbad” [Wojciech Jarosz 2014]




“Sunset from Rigi Kaltbad”

[Wojciech Jarosz 2014]


“Camogli Lighthouse” [Wojciech Jarosz 2012]

“Camogli Lighthouse”

[Wojciech Jarosz 2012]

“Florence”

[Wojciech Jarosz 2011]

“Florence” [Wojciech Jarosz 2011]

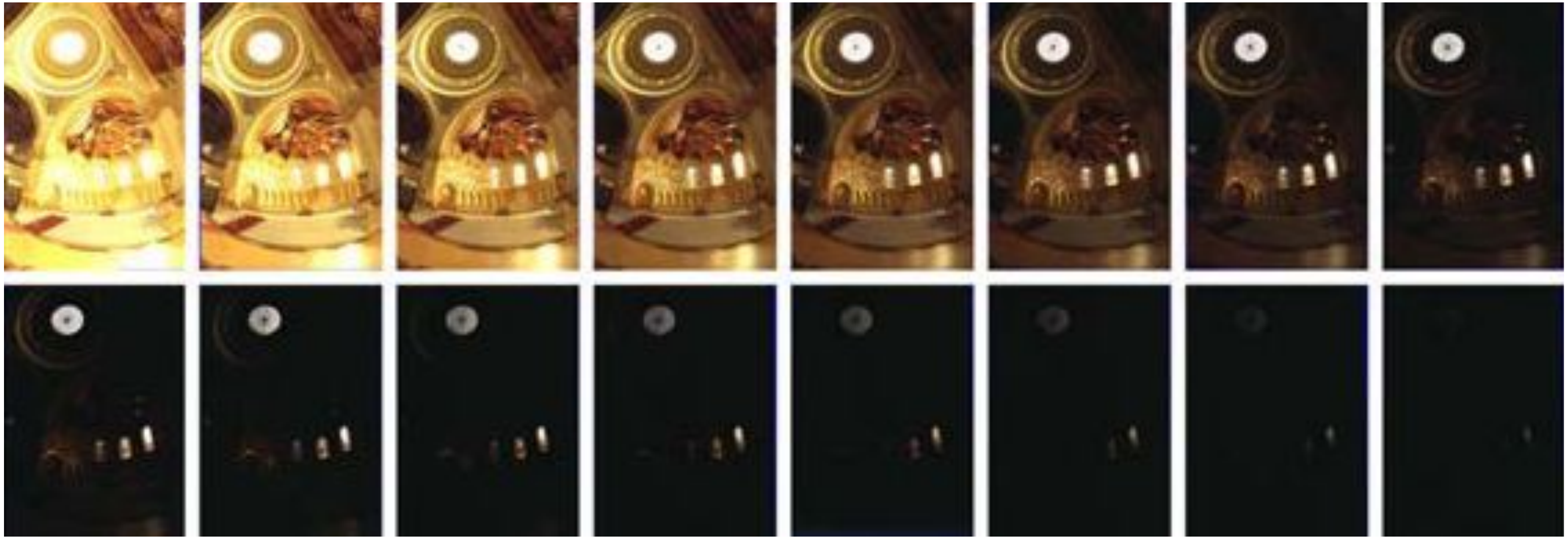
“Matterhorn and Riffelsee” [Wojciech Jarosz 2010]

“Matterhorn and Riffelsee”

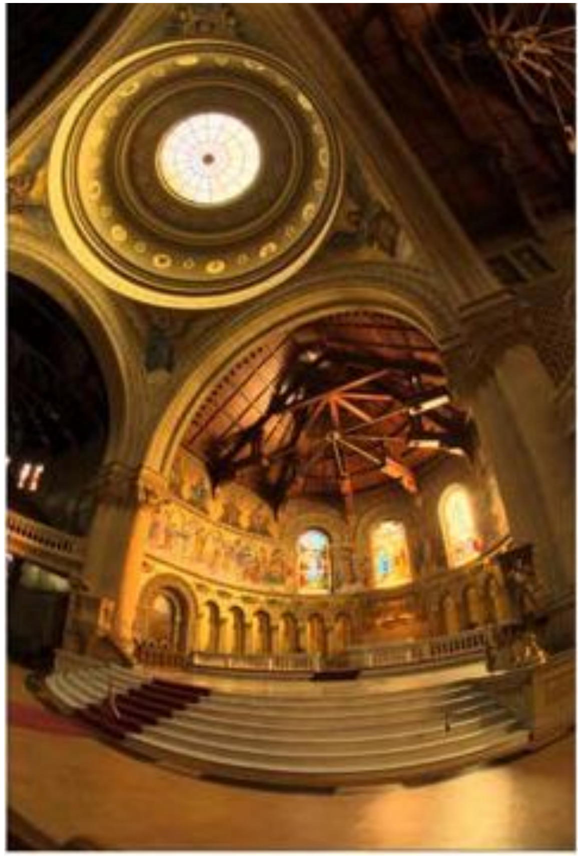
[Wojciech Jarosz 2010]

Key idea

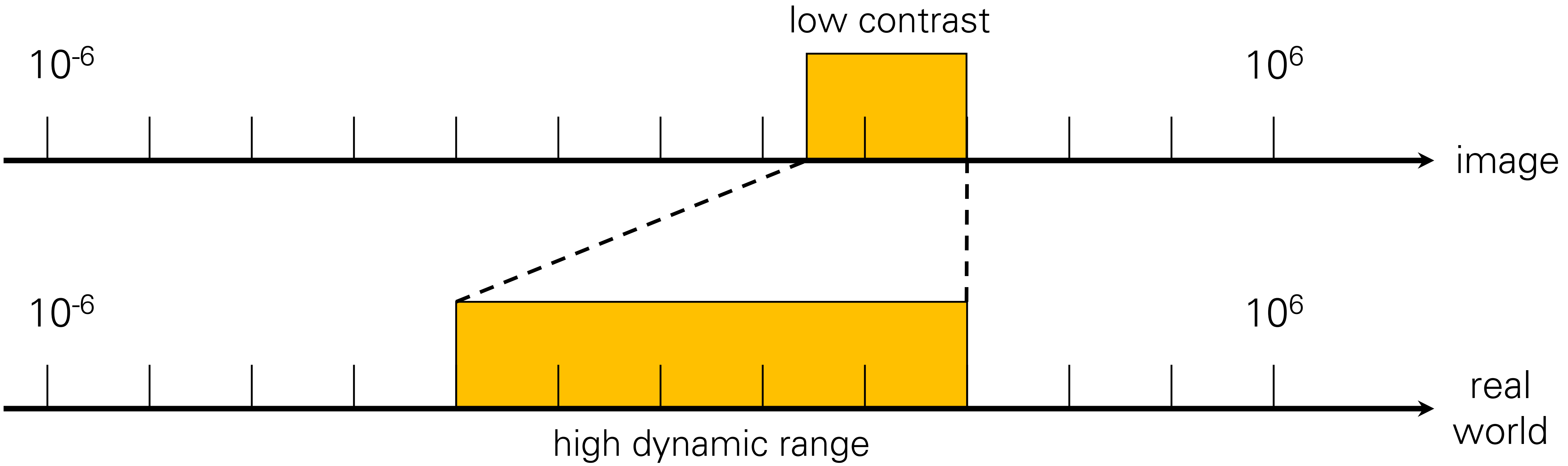
1. Exposure bracketing: Capture multiple LDR images at different exposures



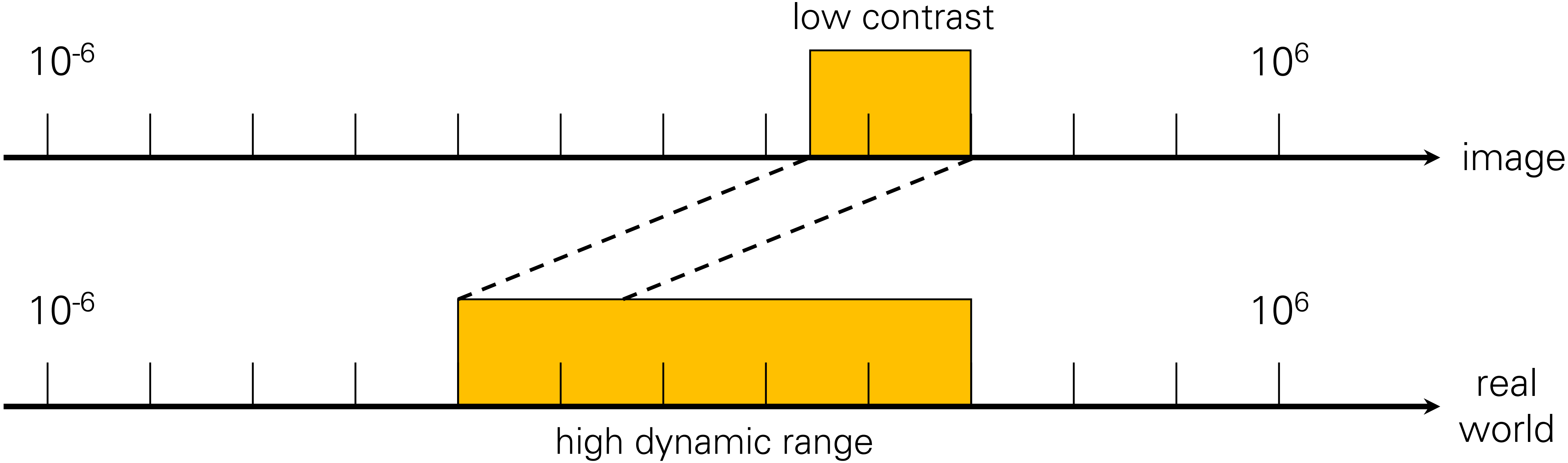
2. Merging: Combine them into a single HDR image



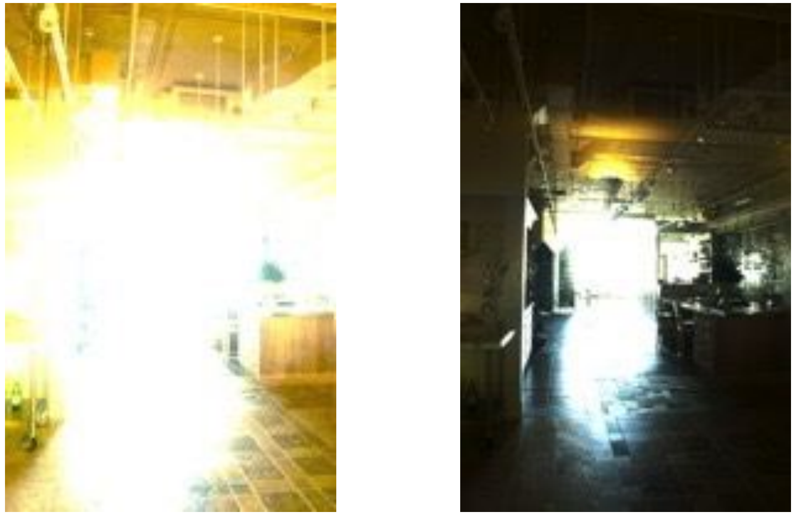
Multiple exposure photography

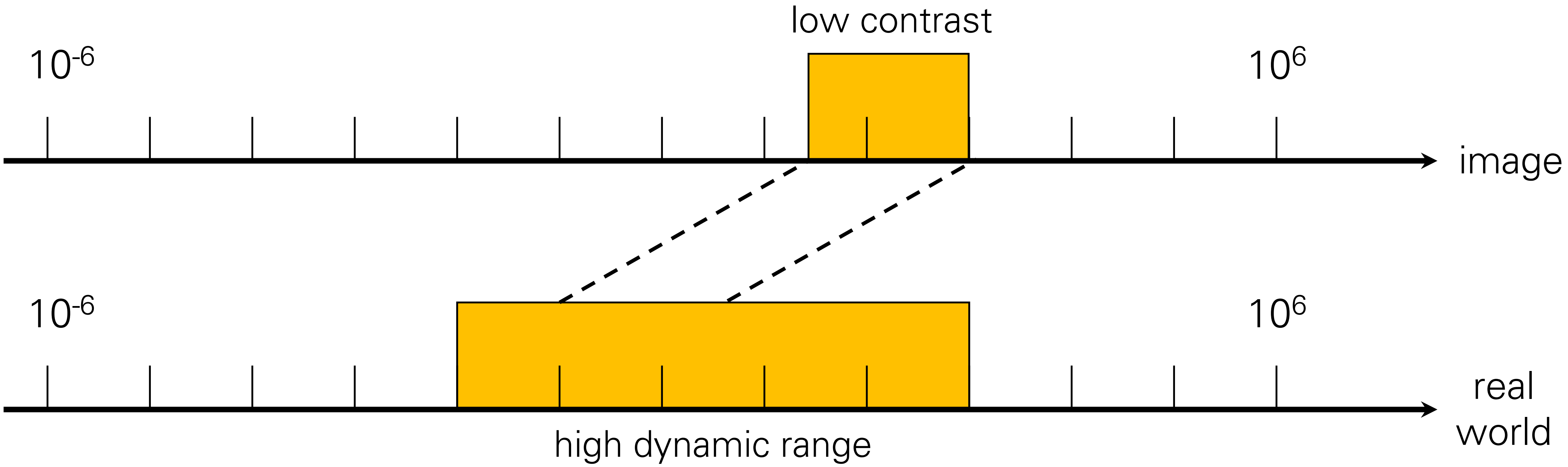


Multiple exposure photography

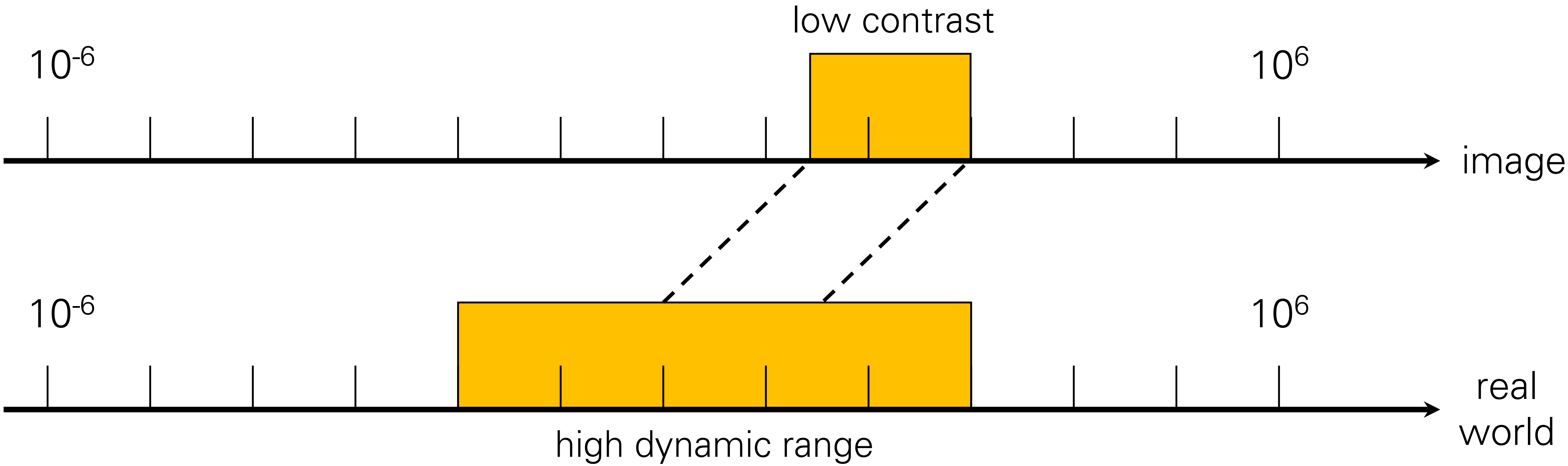


Multiple exposure photography

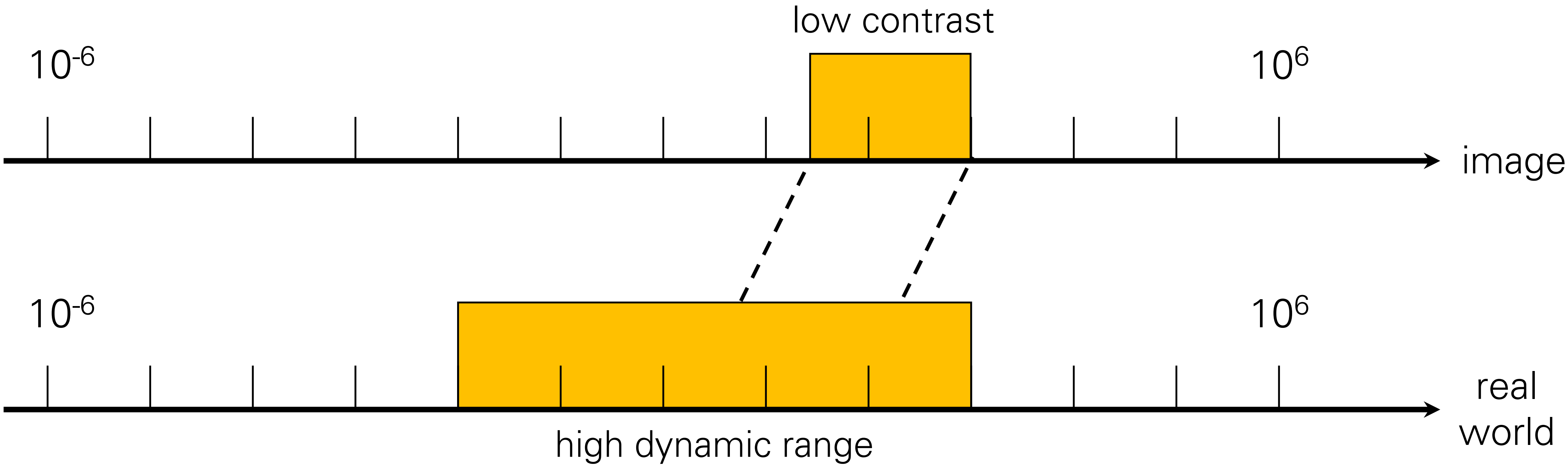




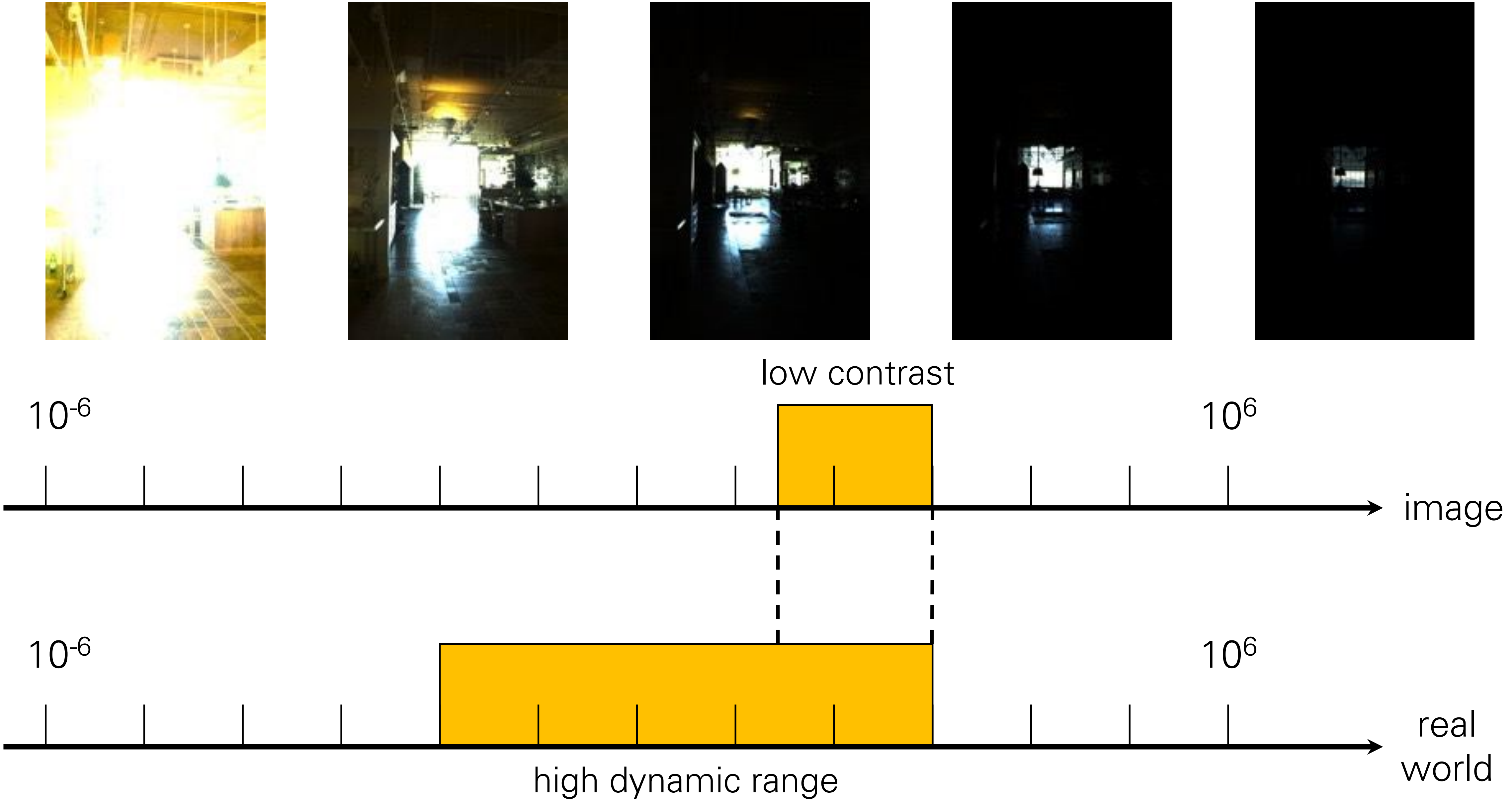
Multiple exposure photography



Multiple exposure photography

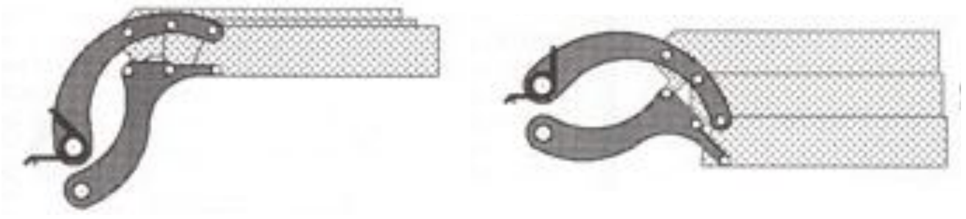


Multiple exposure photography

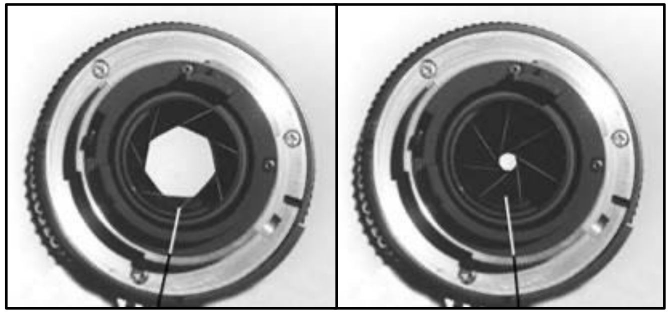


Ways to vary exposure

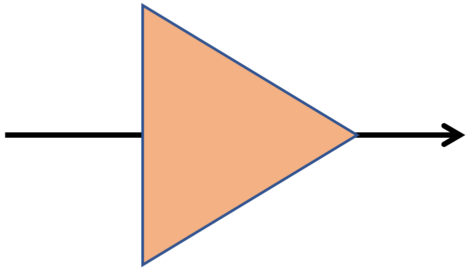
1. Shutter speed



2. F-stop (aperture, iris)



3. ISO



4. Neutral density (ND) filters

Pros and cons of each for HDR?

Ways to vary exposure

1. Shutter speed
   - Range: about 30 sec to 1/4000 sec (6 orders of magnitude)
   - Pros: repeatable, linear
   - Cons: noise and motion blur for long exposure
2. F-stop (aperture, iris)
   - Range: about f/0.98 to f/22 (3 orders of magnitude)
   - Pros: fully optical, no noise
   - Cons: changes depth of field
3. ISO
   - Range: about 100 to 1600 (1.5 orders of magnitude)
   - Pros: no movement at all
   - Cons: noise
4. Neutral density (ND) filters
   - Range: up to 6 densities (6 orders of magnitude)
   - Pros: works with strobe/flash
   - Cons: not perfectly neutral (color shift), extra glass (interreflections, aberrations), need to touch camera (shake)

Exposure bracketing with shutter speed

Note: shutter times usually obey a power series – each "stop" is a factor of 2

1/4, 1/8, 1/15, 1/30, 1/60, 1/125, 1/250, 1/500, 1/1000 sec

usually really is

1/4, 1/8, 1/16, 1/32, 1/64, 1/128, 1/256, 1/512, 1/1024 sec

Questions:

1. How many exposures?
2. What exposures?

Exposure bracketing with shutter speed

Note: shutter times usually obey a power series – each “stop” is a factor of 2

1/4, 1/8, 1/15, 1/30, 1/60, 1/125, 1/250, 1/500, 1/1000 sec

usually really is

1/4, 1/8, 1/16, 1/32, 1/64, 1/128, 1/256, 1/512, 1/1024 sec

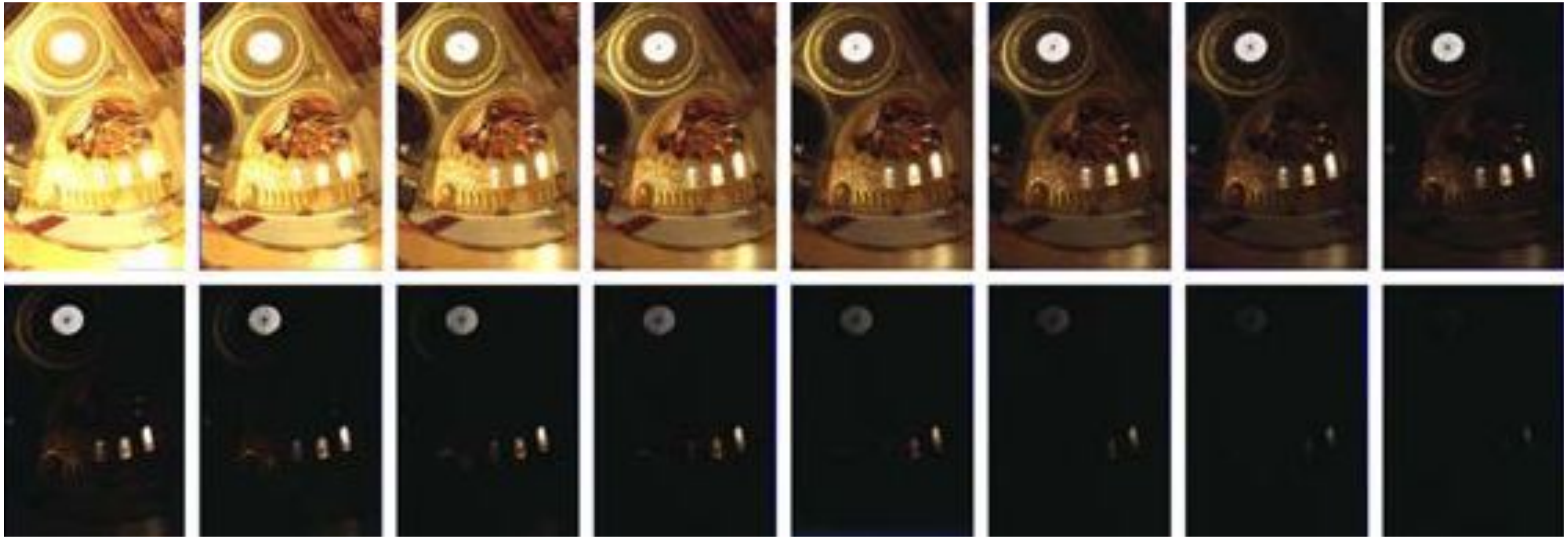
Questions:

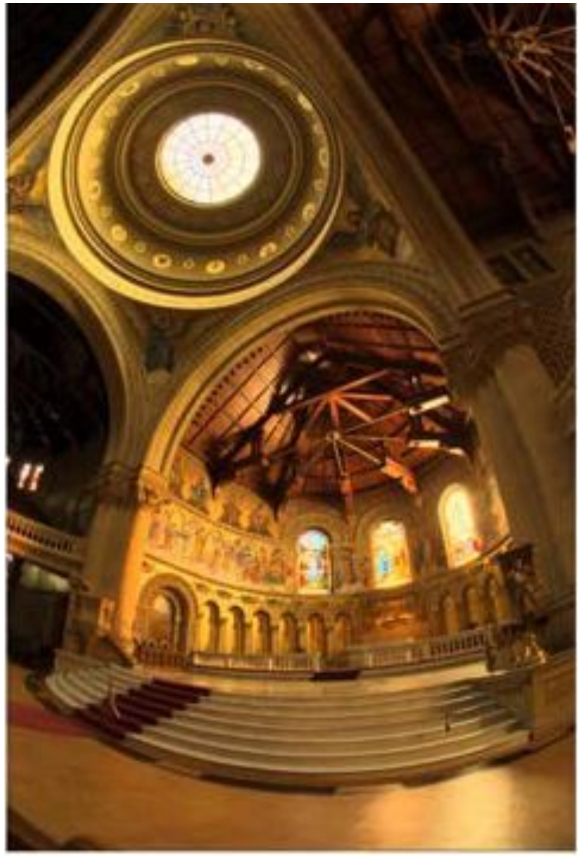
1. How many exposures?
2. What exposures?

Answer: Depends on the scene, but a good default is 5 exposures, the metered exposure and +/- 2 stops around that.

Key idea

1. Exposure bracketing: Capture multiple LDR images at different exposures

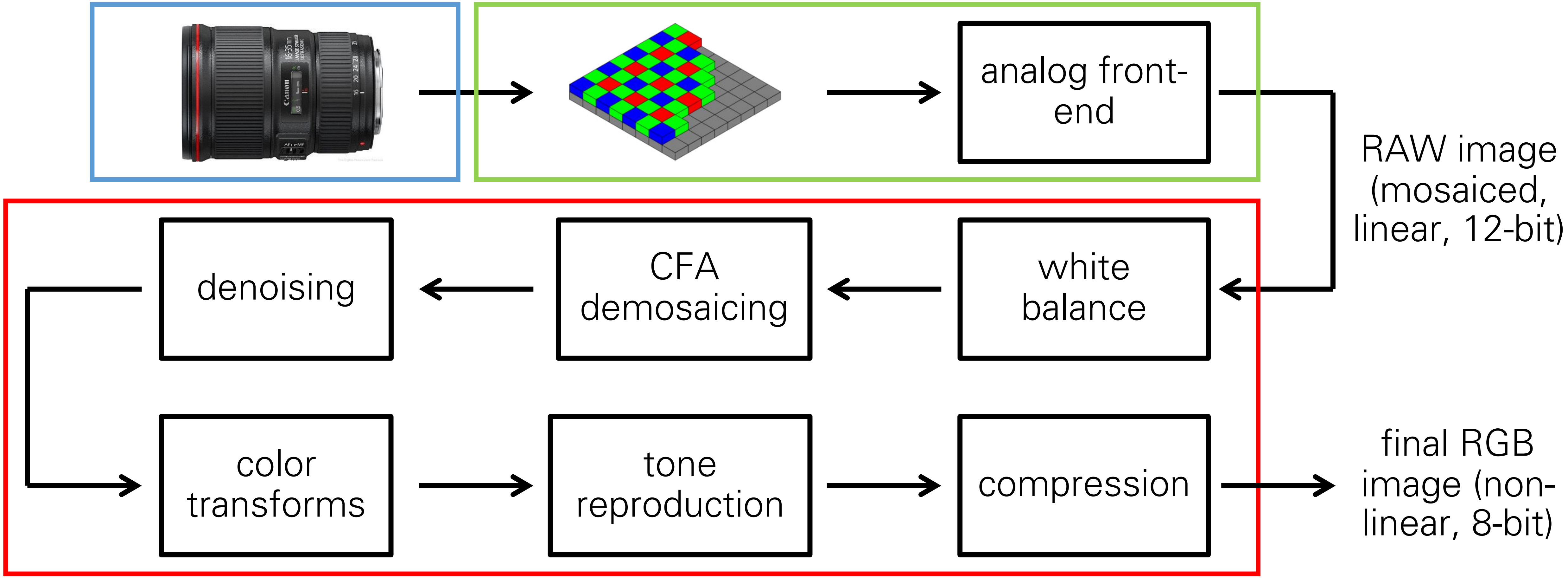




2. Merging: Combine them into a single HDR image

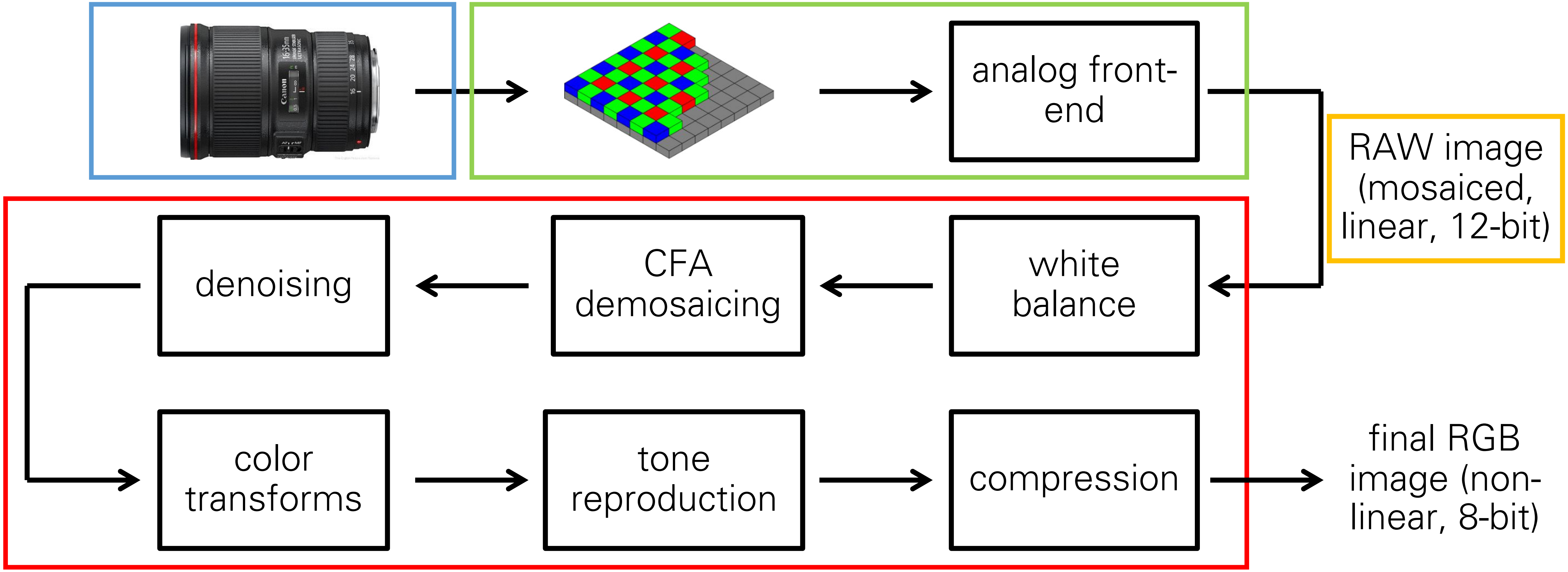
The image processing pipeline

The sequence of image processing operations applied by the camera's image signal processor (ISP) to convert a RAW image into a "conventional" image.



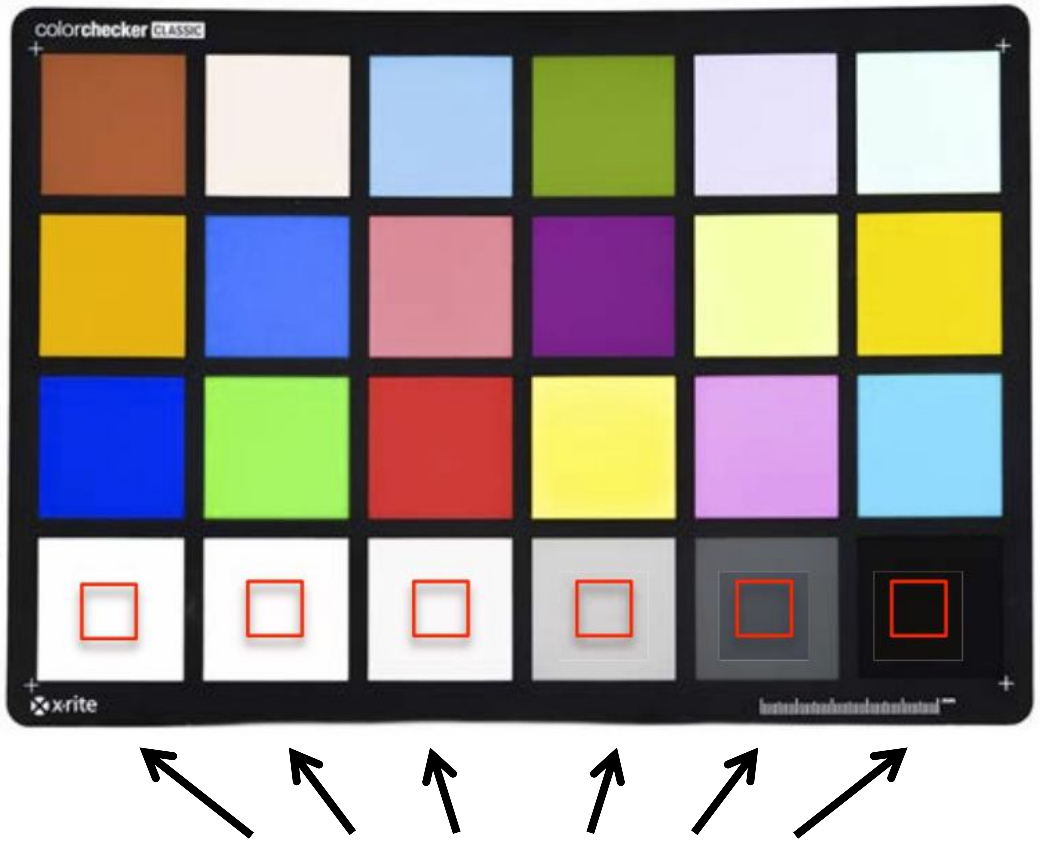
The image processing pipeline

The sequence of image processing operations applied by the camera's image signal processor (ISP) to convert a RAW image into a "conventional" image.

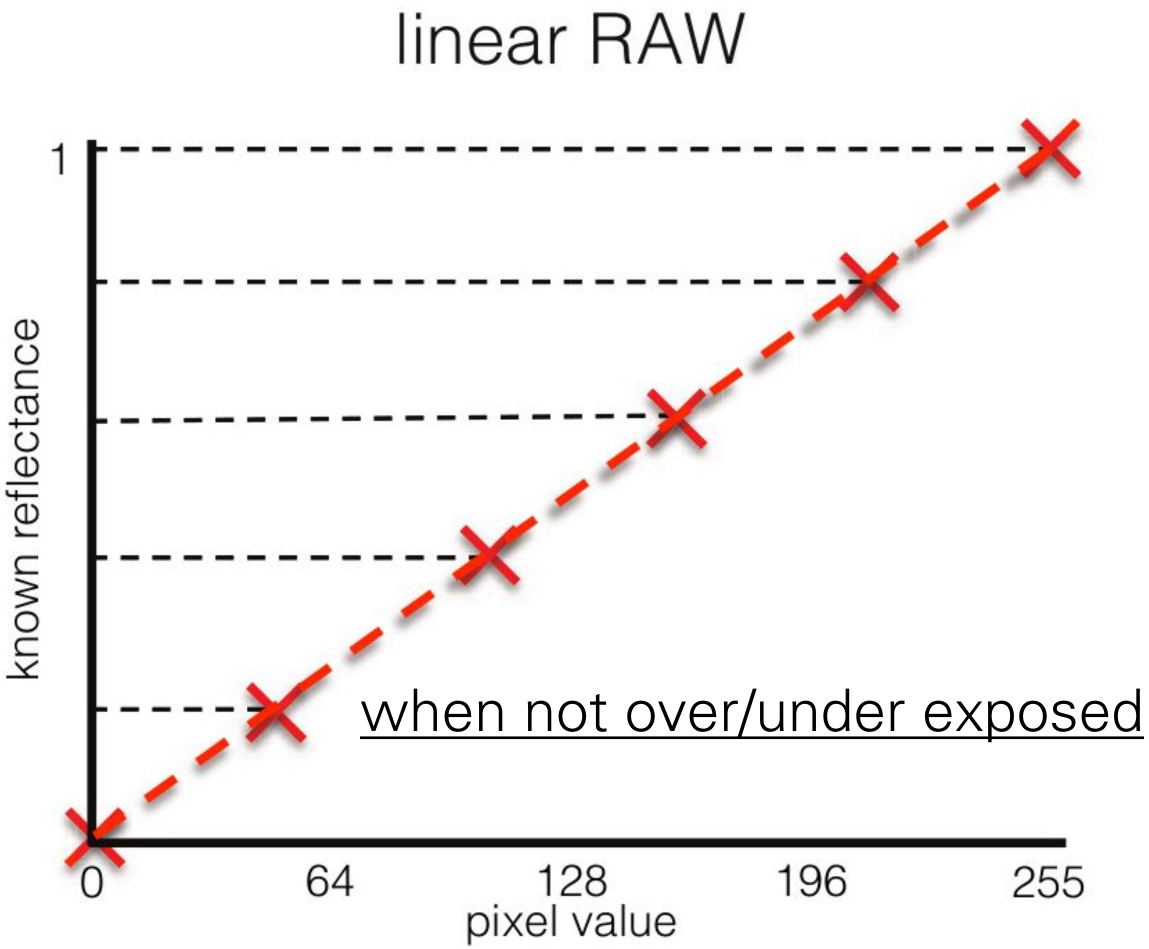


RAW images have a linear response curve

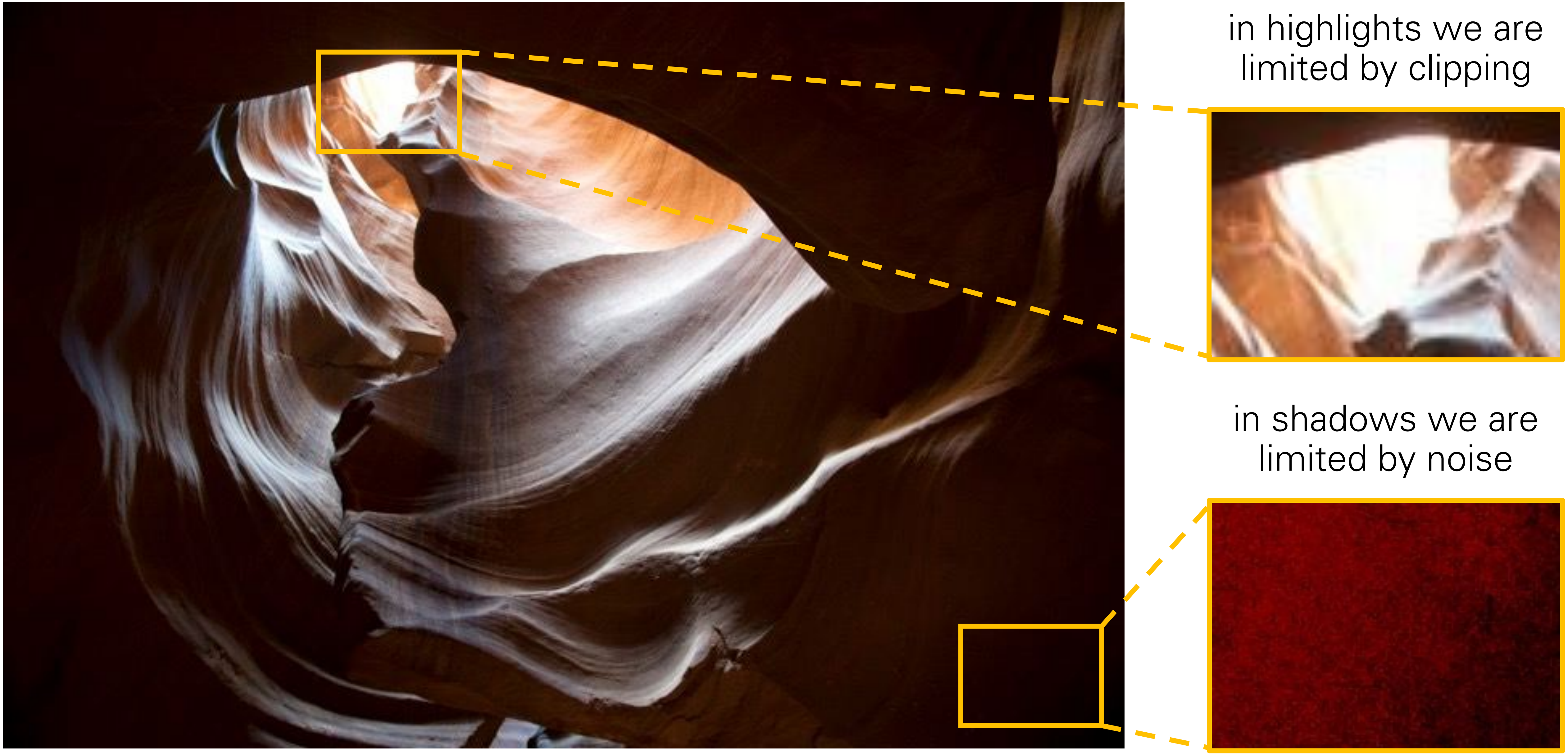
<u>Colorchecker:</u> Great tool for radiometric and color calibration.



Patches at bottom row have log-reflectance that increases linearly.



Over/under exposure



in highlights we are limited by clipping

in shadows we are limited by noise

RAW (linear) image formation model

Real scene flux for image pixel (x,y): $\Phi(x, y)$

Exposure time:

t_5 t_4 t_3 t_2 t_1

What is an expression for the image $I_{linear}(x,y)$ as a function of $\Phi(x,y)$?

RAW (linear) image formation model

Real scene flux for image pixel (x,y): $\Phi(x, y)$

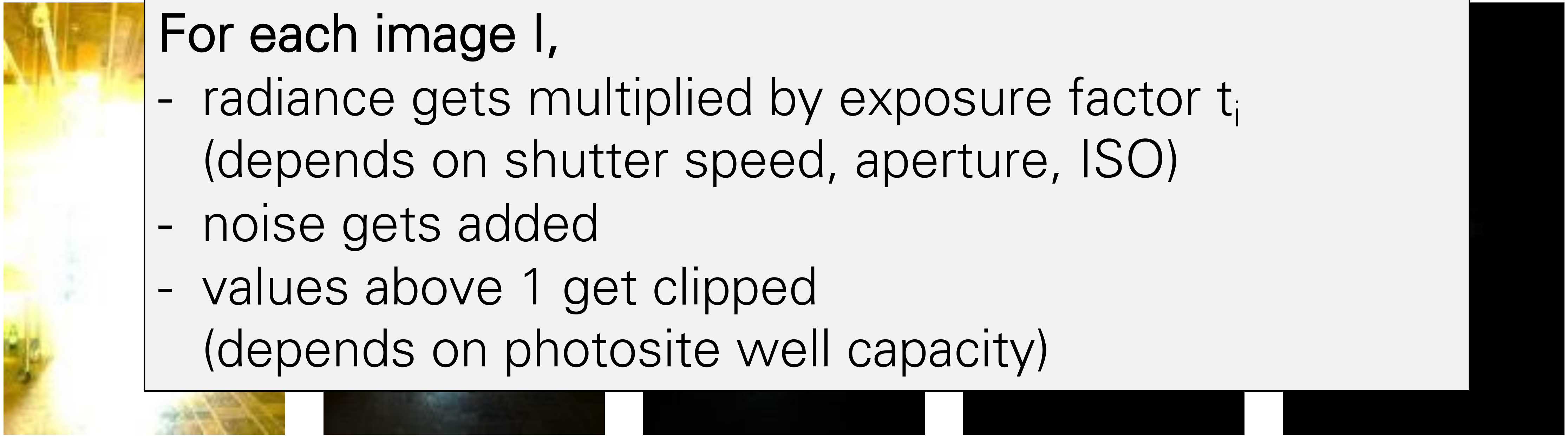
Exposure time:

t

Scene radiance $\Phi(x,y)$ reaches the sensor at a pixel x, y

For each image I,

- radiance gets multiplied by exposure factor t_i (depends on shutter speed, aperture, ISO)
- noise gets added
- values above 1 get clipped (depends on photosite well capacity)



What is an expression for the image $I_{linear}(x,y)$ as a function of $\Phi(x,y)$?

RAW (linear) image formation model

Real scene flux for image pixel (x,y): $\Phi(x, y)$

Exposure time:

t_5 t_4 t_3 t_2 t_1

What is an expression for the image $I_{linear}(x,y)$ as a function of $\Phi(x,y)$?

$$I_{linear}(x,y) = \text{clip}[\ t_i \cdot \Phi(x,y) + \text{noise}\ ]$$

RAW (linear) image formation model

Real scene flux for image pixel (x,y): $\Phi(x, y)$

Exposure time:

t_5 t_4 t_3 t_2 t_1

What is an expression for the image $I_{linear}(x,y)$ as a function of $\Phi(x,y)$?

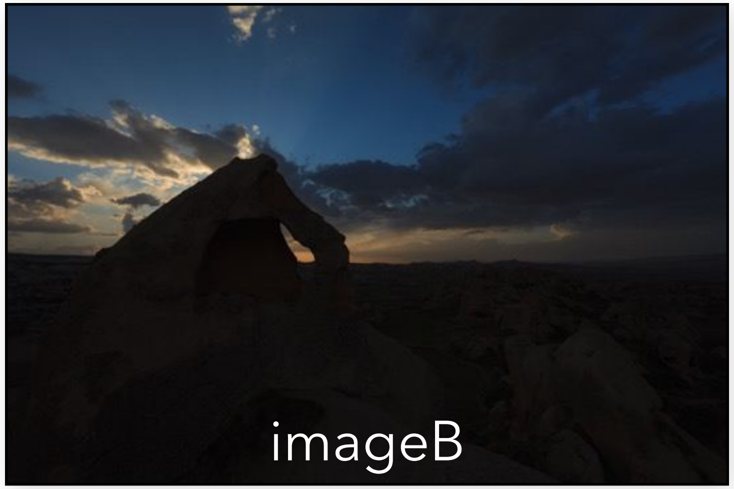
$$I_{linear}(x,y) = \text{clip}[\ t_i \cdot \Phi(x,y) + \text{noise}\ ]$$

How would you merge these images into an HDR one?

2-image example

Simple in principle:

- imageA = 1/30th second ("brighter" image)
- imageB = 1/120th second ("darker" image)
- imageHDR = average(4·imageB, remove-clipped(imageA))
- assumes images have been linearized



Merging RAW (linear) exposure stacks

For each pixel:

1. Find "valid" images
2. Weight valid pixel values appropriately
3. Form a new pixel value as the weighted average of valid pixel values

How would you implement steps 1-2?

t_5

t_4

t_3

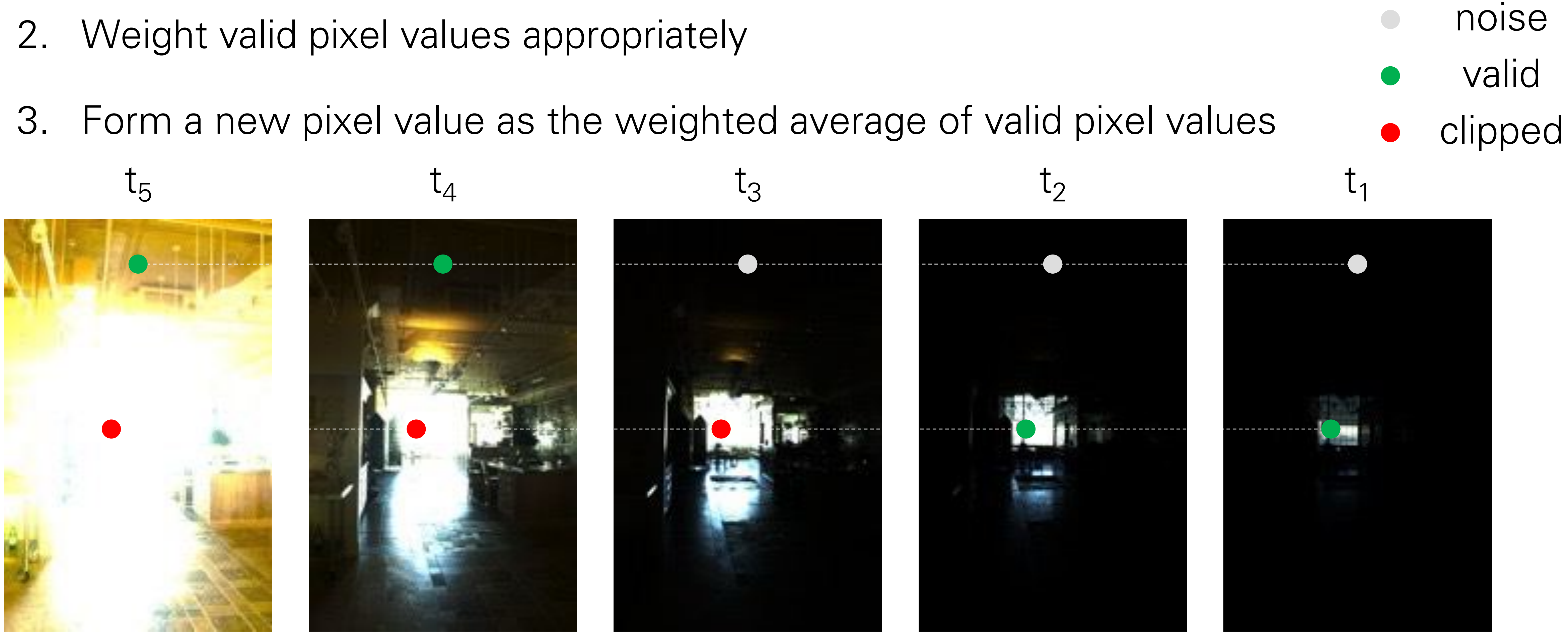
t_2

t_1

Merging RAW (linear) exposure stacks

For each pixel:

1. Find "valid" images ⟵ (noise) $0.05 < \text{pixel} < 0.95$ (clipping)
2. Weight valid pixel values appropriately
3. Form a new pixel value as the weighted average of valid pixel values



Merging RAW (linear) exposure stacks

For each pixel:

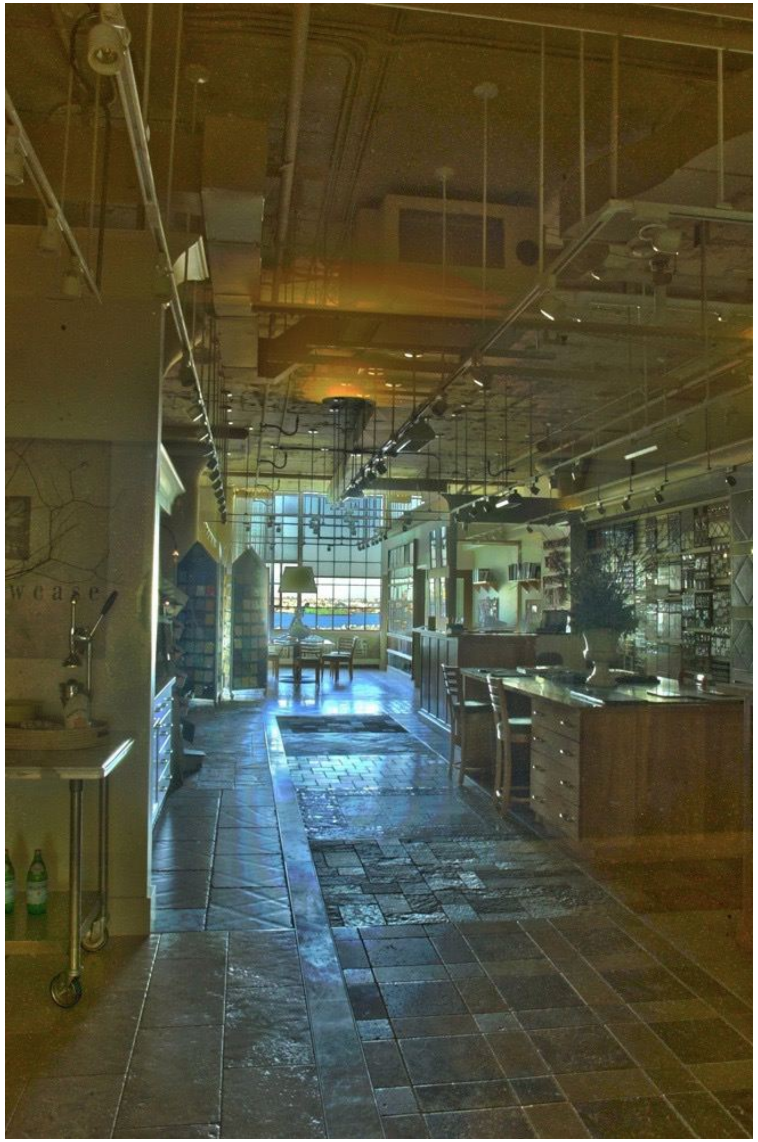
1. Find "valid" images ⟵ (noise) 0.05 < pixel < 0.95 (clipping)

2. Weight valid pixel values appropriately ⟵ (pixel value) / t_i

3. Form a new pixel value as the weighted average of valid pixel values

t_5 t_4 t_3 t_2 t_1

Merging result (after tonemapping)

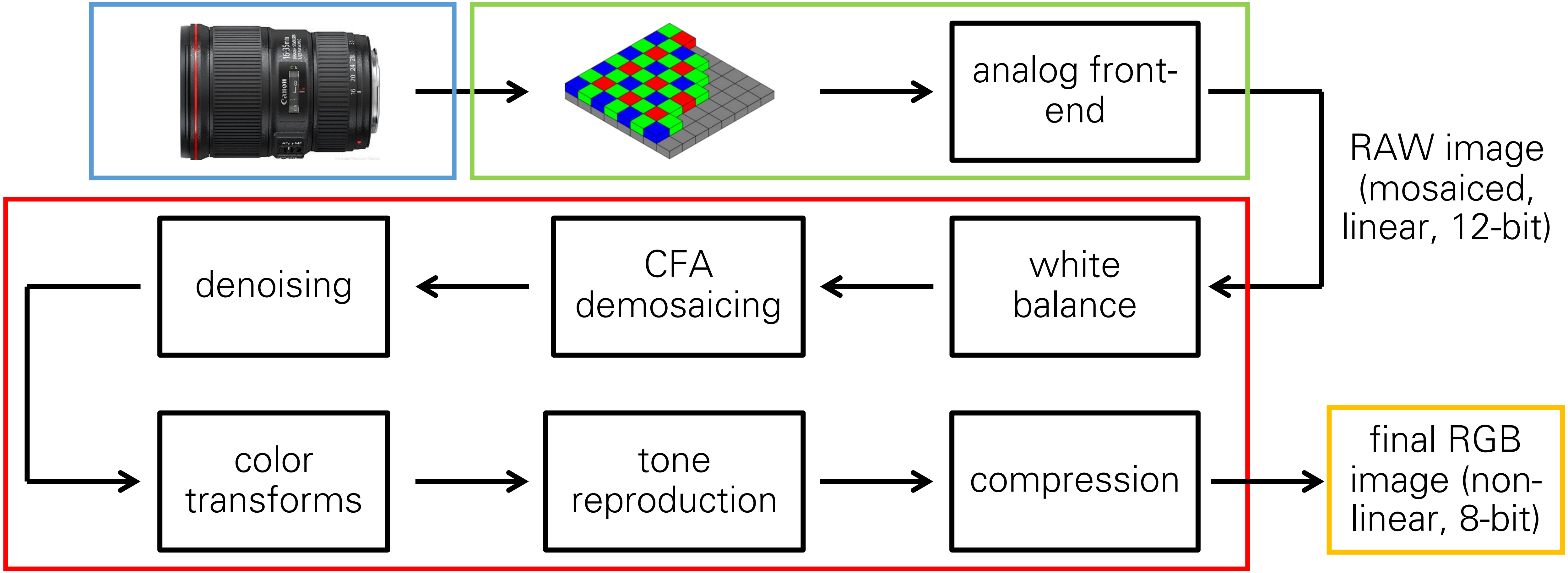


What if I cannot use raw?

Radiometric calibration

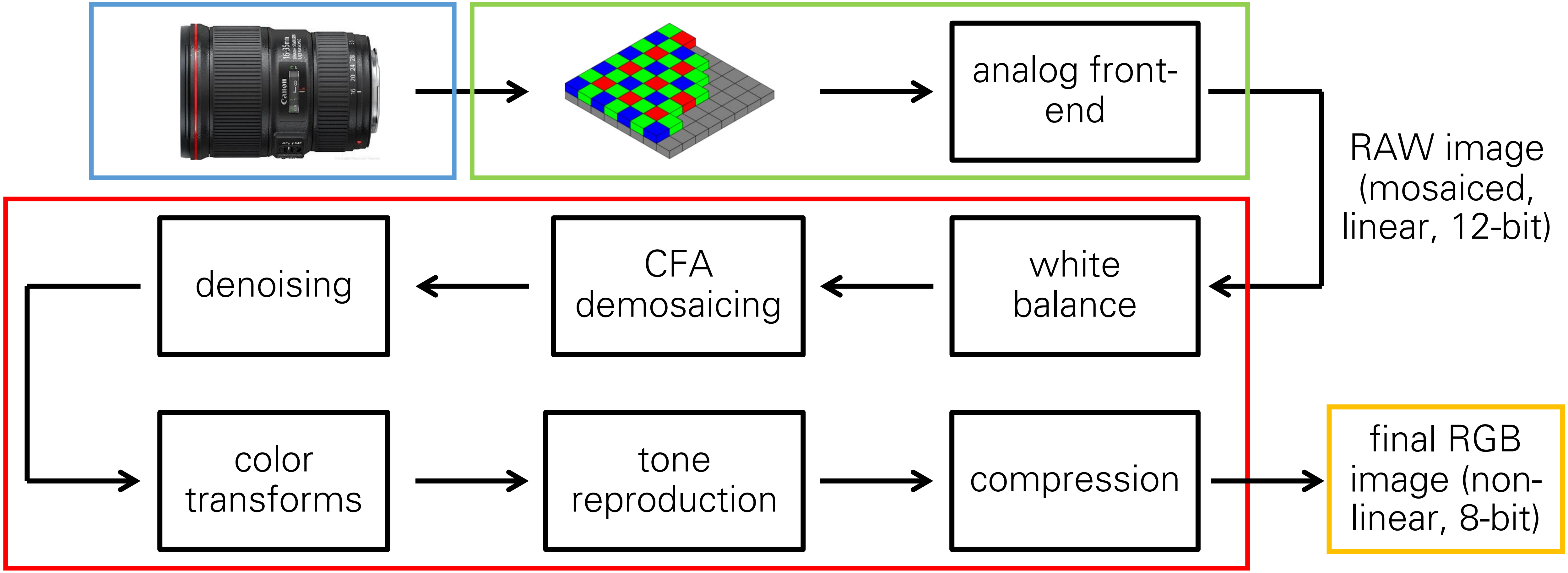
The image processing pipeline

- Can you foresee any problem when we switch from RAW to rendered images?



The image processing pipeline

- Can you foresee any problem when we switch from RAW to rendered images?
- How do we deal with the nonlinearities?



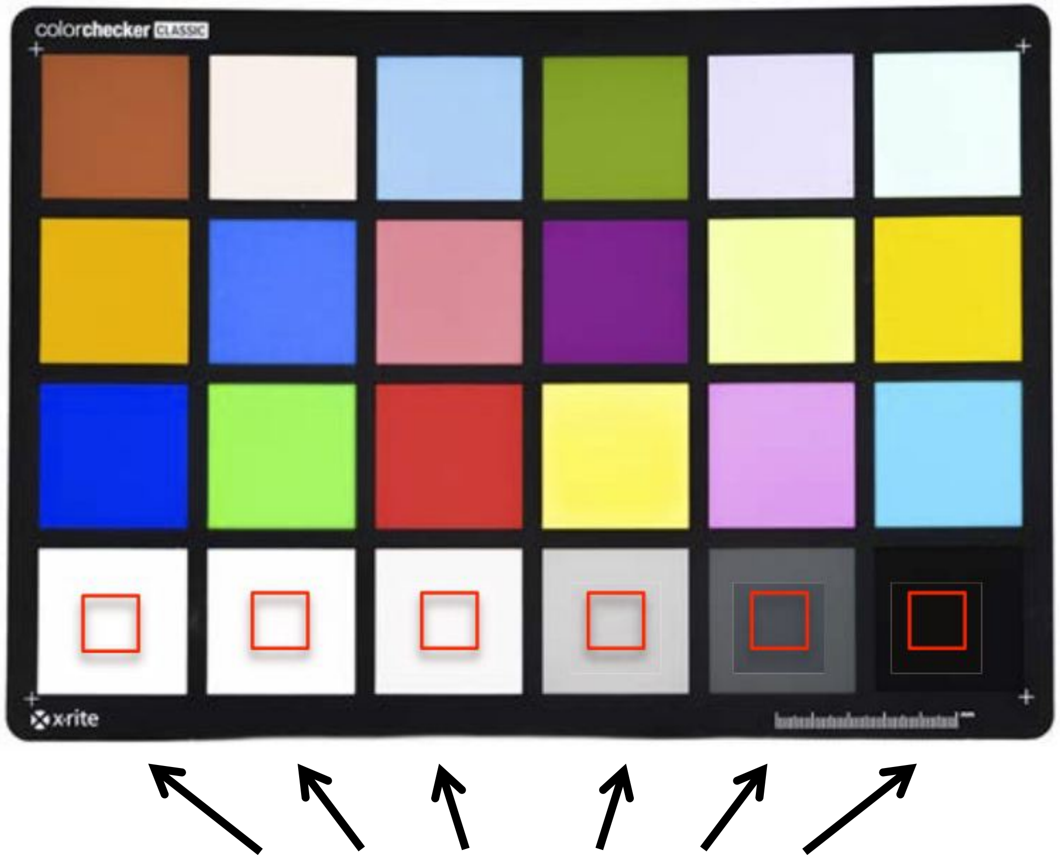
Radiometric calibration

The process of measuring the camera's response curve. Can be done in three ways:

- Take images of scenes with different flux while keeping exposure the same.
- Take images under different exposures while keeping flux the same.
- Take images of scenes with different flux and under different exposures.

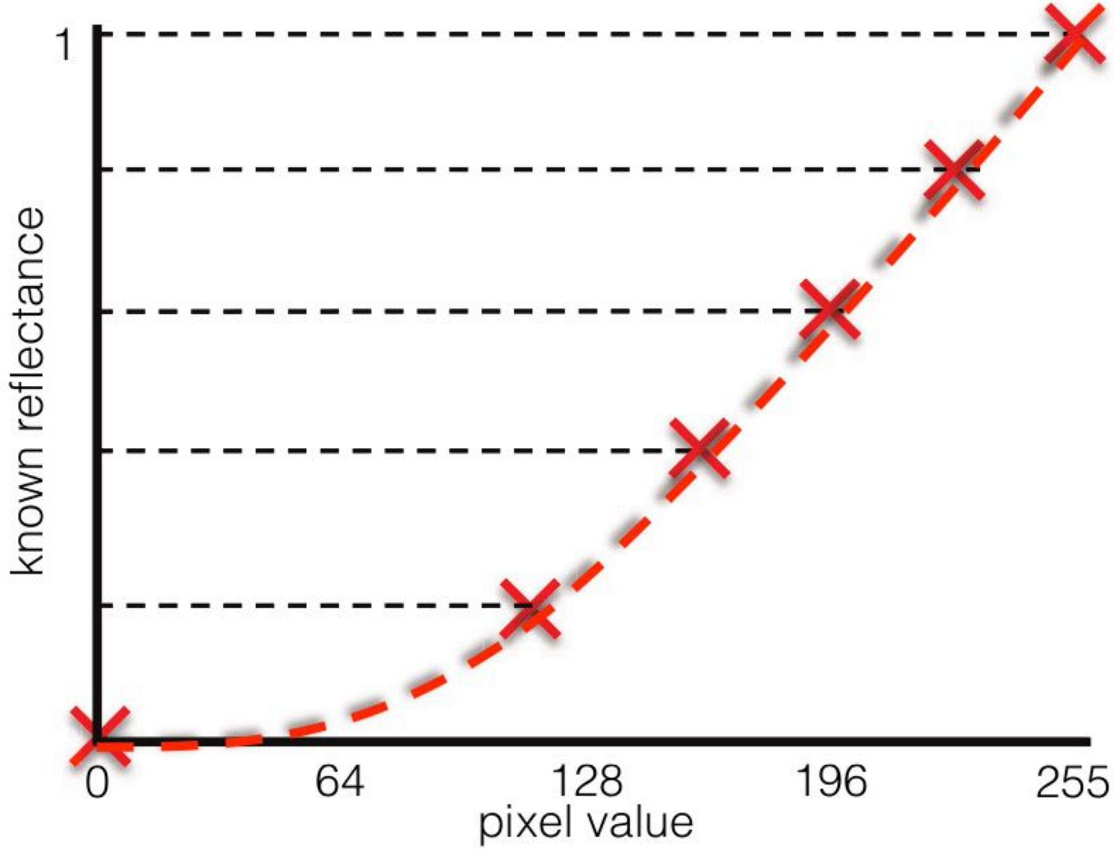
Same camera exposure, varying scene flux

Colorchecker: Great tool for radiometric and color calibration.



Patches at bottom row have log-reflectance that increases linearly.

e.g. JPEG

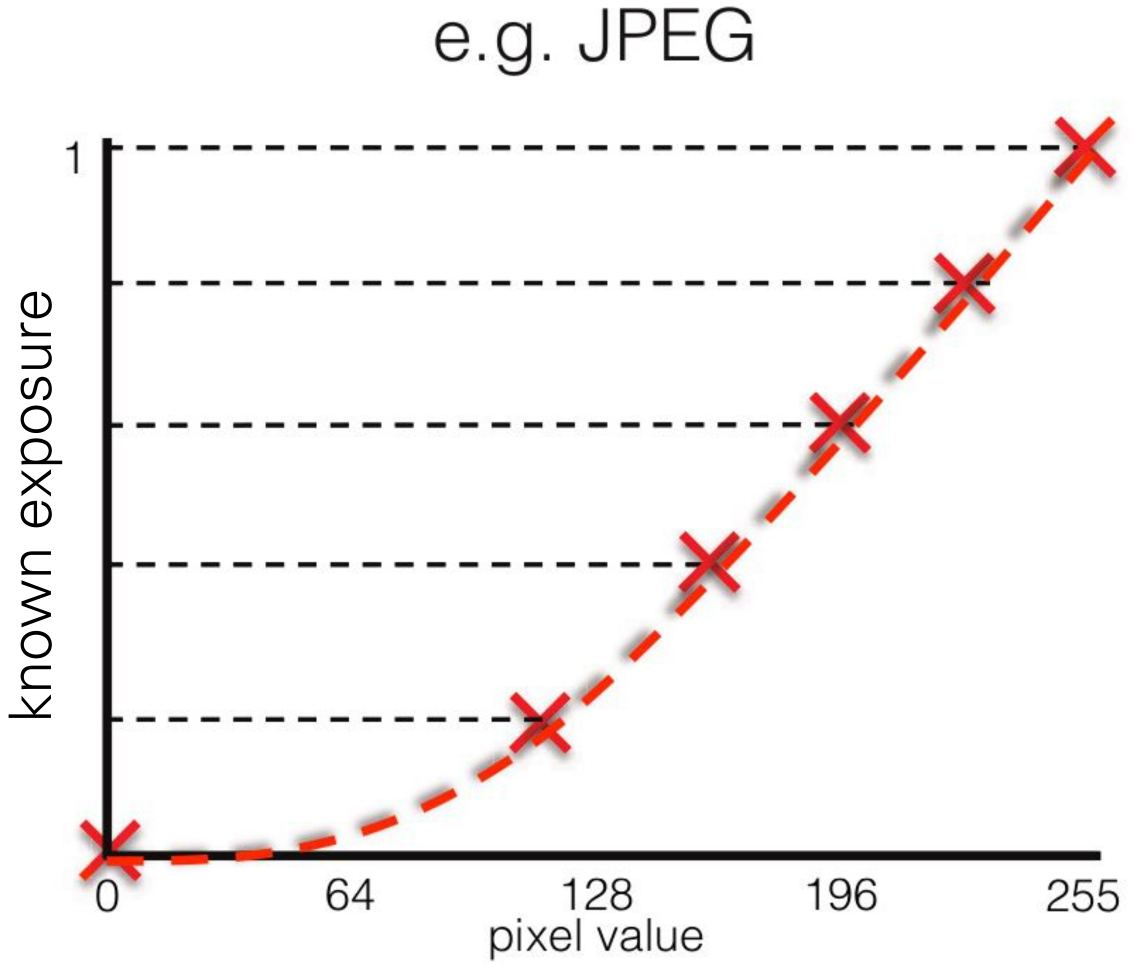


Different values correspond to patches of increasing reflected flux.

Same scene flux, varying camera exposure

<u>White balance card:</u> Great tool for white balancing and radiometric calibration.

All points on (the white part of) the target have the same reflectance.

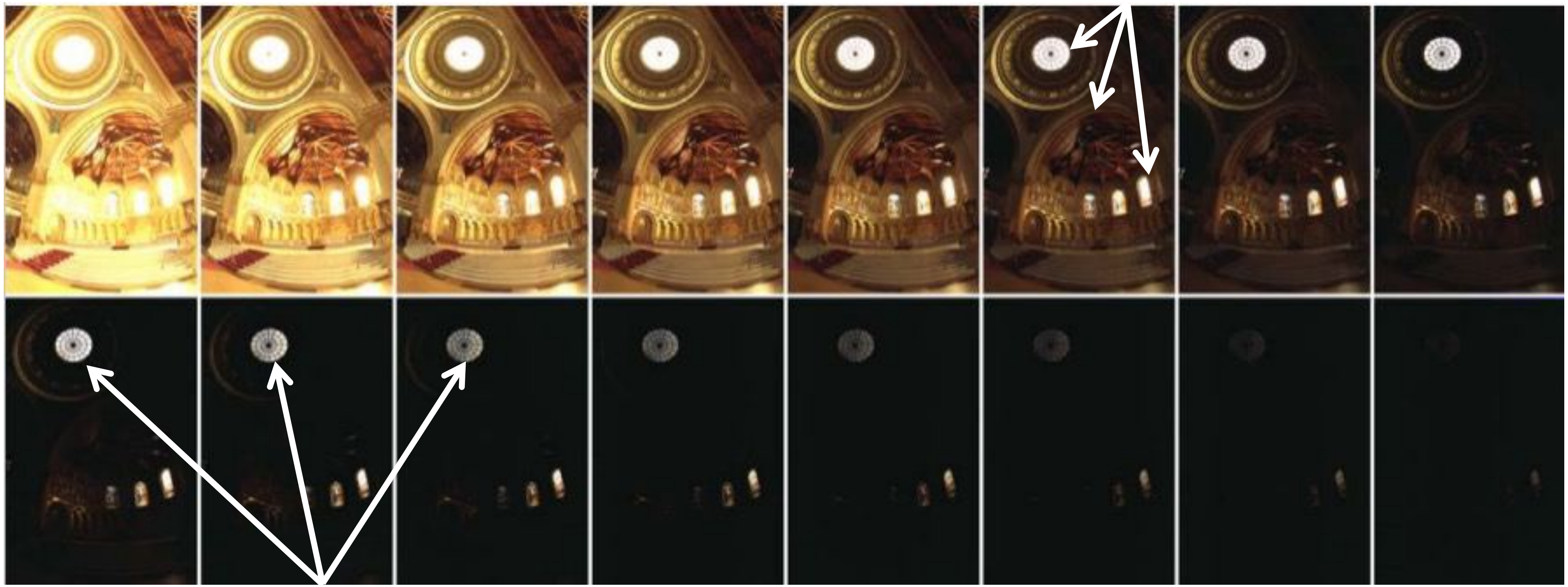


Different values correspond to images taken under increasing camera exposure.

Varying both scene flux and camera exposure

You can do this using the LDR exposure stack itself.

Different scene flux, same camera exposure

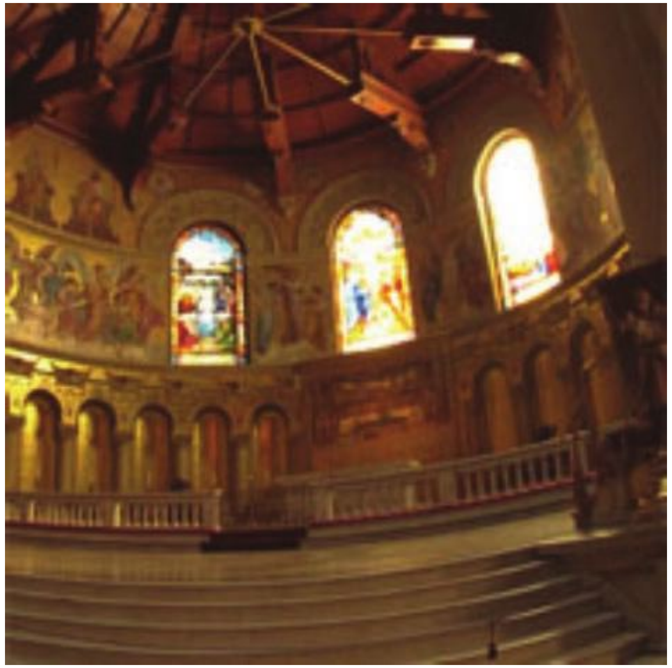


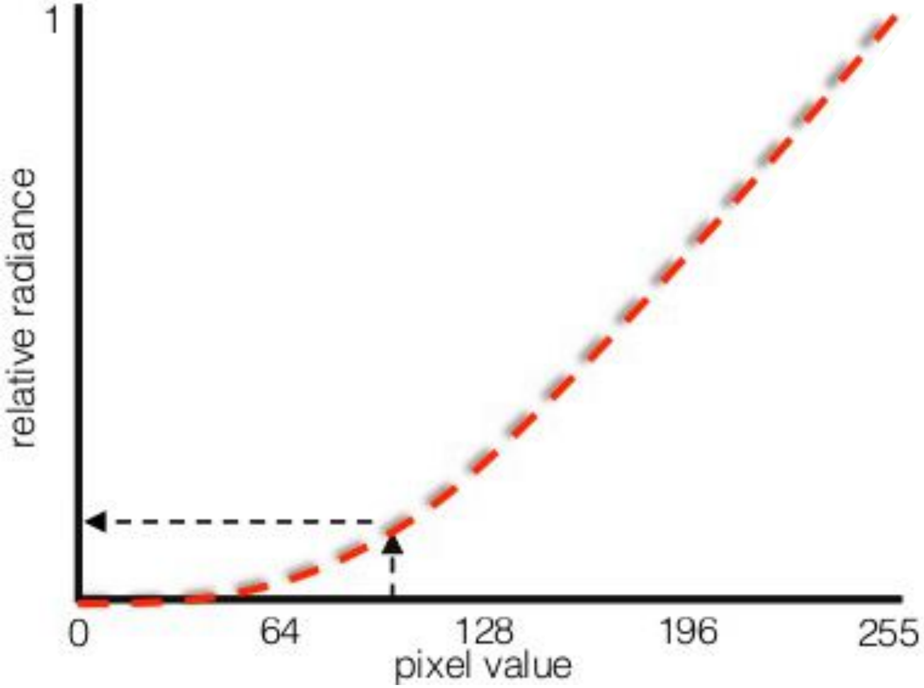
Same scene flux, different camera exposure

Non-linear image formation model

Real scene flux for image pixel (x,y): $\Phi(x, y)$

Exposure time: t_i





$I_{linear}(x,y) = \text{clip}[\ t_i \cdot \Phi(x,y) + \text{noise}\ ]$

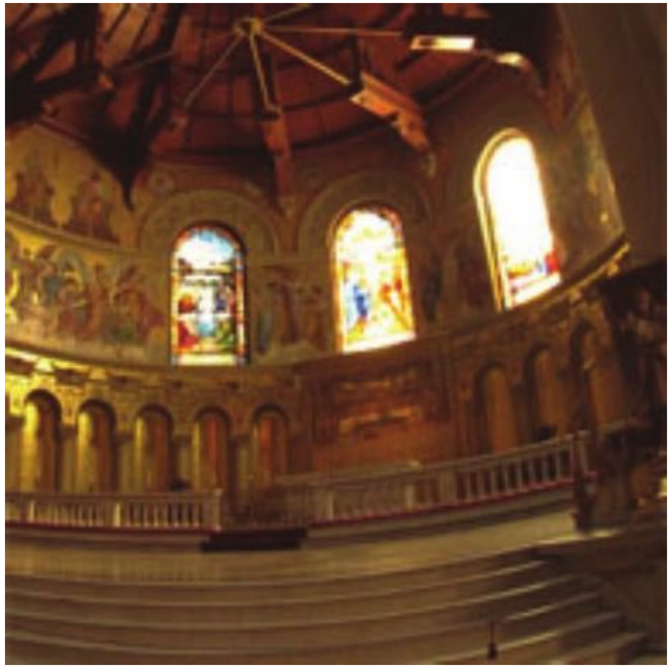
$I_{non\text{-}linear}(x,y) = f[\ I_{linear}(x,y)\ ]$

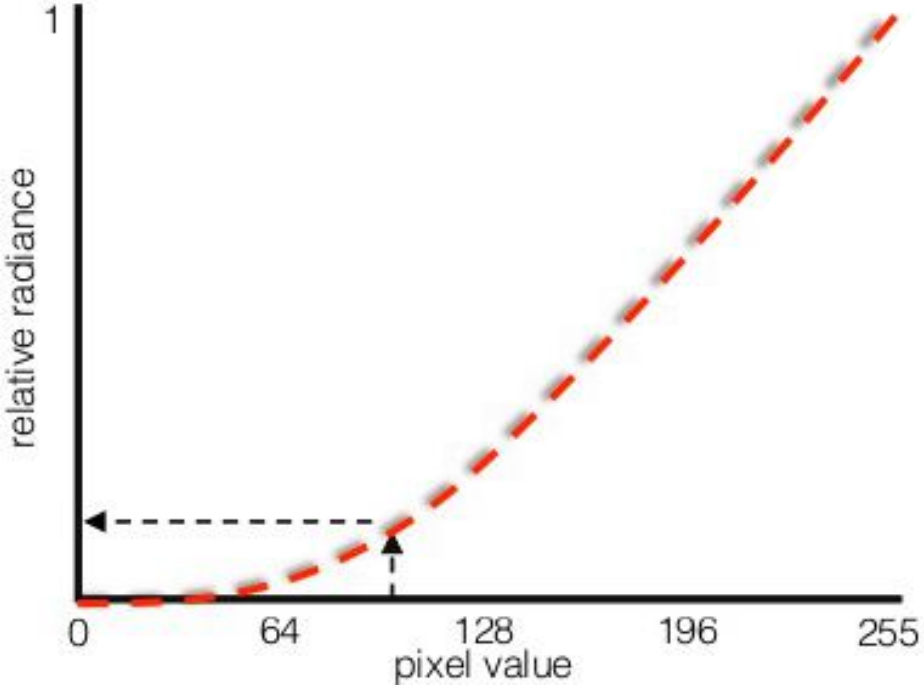
How would you merge the non-linear images into an HDR one?

Non-linear image formation model

Real scene flux for image pixel (x,y): $\Phi(x, y)$

Exposure time: t_i





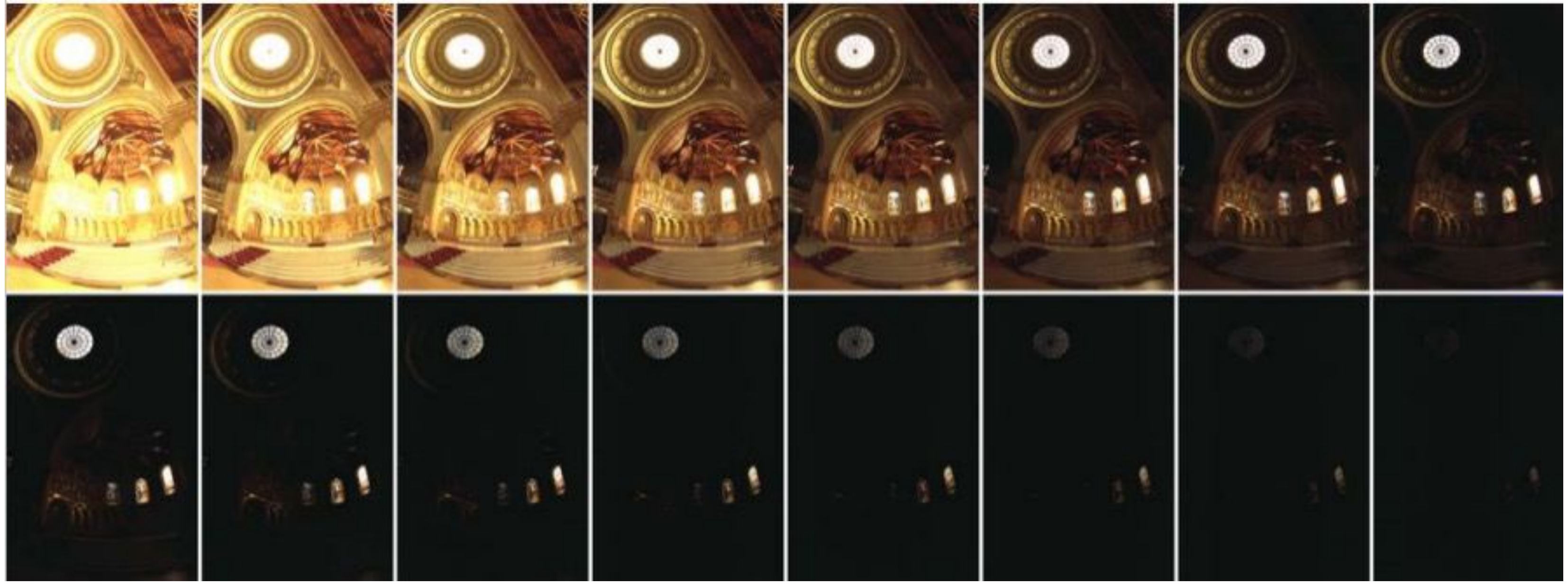
$I_{linear}(x,y) = \text{clip}[\ t_i \cdot \Phi(x,y) + \text{noise}\ ]$

$I_{non\text{-}linear}(x,y) = f[\ I_{linear}(x,y)\ ]$ $\qquad I_{est}(x,y) = f^{-1}[\ I_{non\text{-}linear}(x,y)\ ]$

Use inverse transform to estimate linear image, then proceed as before

Linearization

$I_{non\text{-}linear}(x,y) = f[\ I_{linear}(x,y)\ ]$



$I_{est}(x,y) = f^{-1}[\ I_{non\text{-}linear}(x,y)\ ]$

Merging non-linear exposure stacks

1. Calibrate response curve
2. Linearize images

For each pixel:

3. Find “valid” images ← (noise) 0.05 < pixel < 0.95 (clipping)
4. Weight valid pixel values appropriately ← (pixel value) / t_i
5. Form a new pixel value as the weighted average of valid pixel values

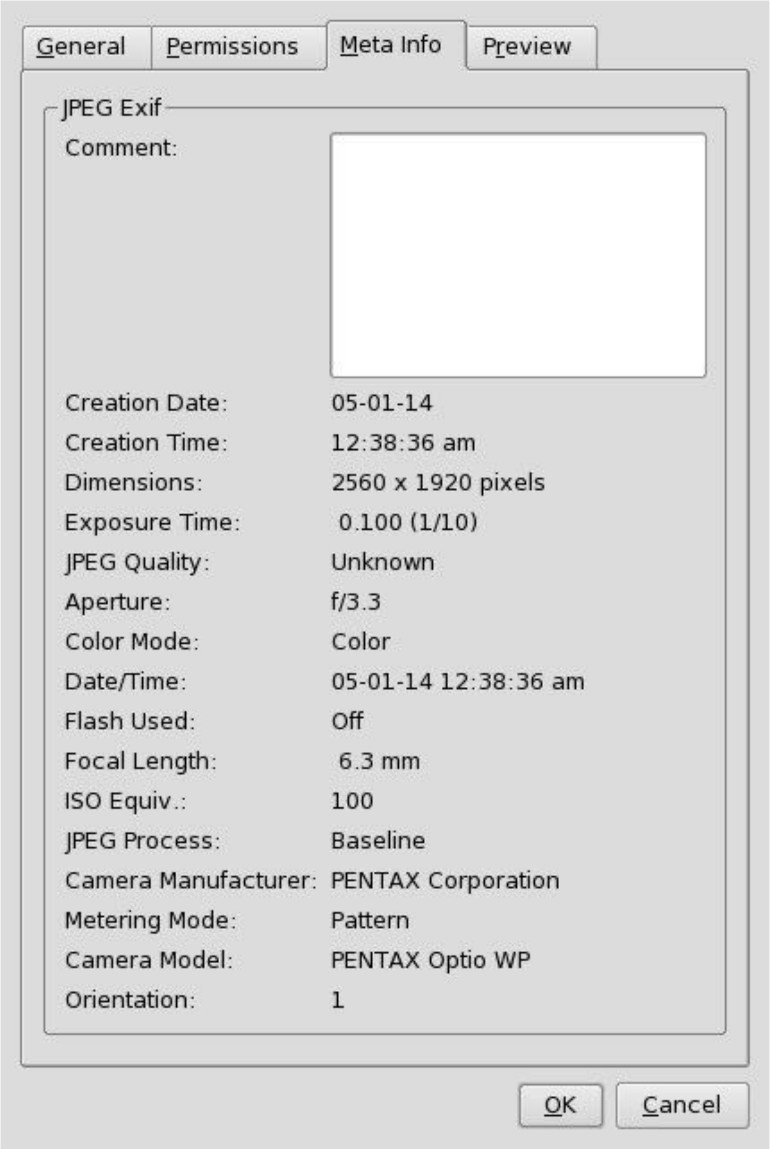
Same steps as in the RAW case.

What if I cannot measure the response curve?

You may find information in the image itself

If you cannot do calibration, take a look at the image's EXIF data (if available).

Often contains information about tone reproduction curve and color space.



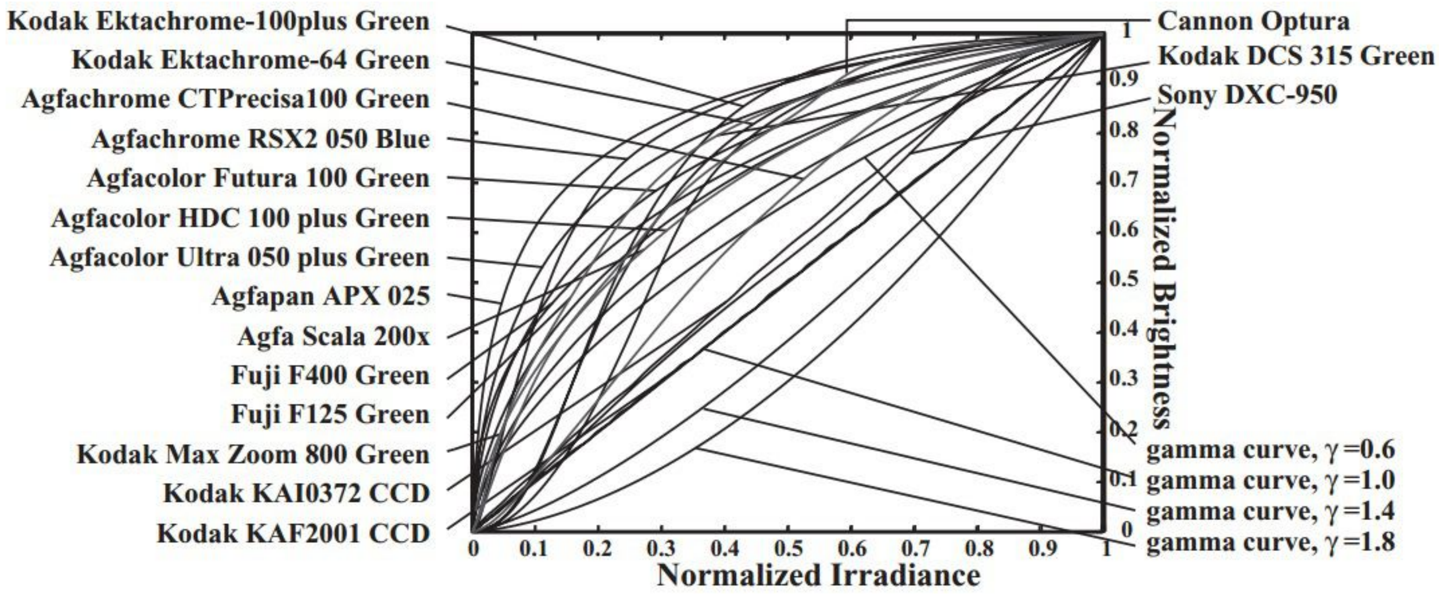
Tone reproduction curves

The exact tone reproduction curve depends on the camera.

- Often well approximated as L^γ, for different values of the power γ ("gamma").
- A good default is $\gamma = 1 / 2.2$.

before gamma

after gamma



If nothing else, take the square of your image to approximately remove effect of tone reproduction curve.

What if I cannot measure the response curve?

- Predict an approximated camera response function from the observed images.

The Approach

- Get pixel values Z_{ij} for image with shutter time Δt_j (i^{th} pixel location, j^{th} image)
- Exposure is radiance integrated over time:

$$E_{ij} = R_i \cdot \Delta t_j \Rightarrow \ln E_{ij} = \ln R_i + \ln \Delta t_j$$

- To recover radiance R_i, we must map pixel values to log exposure: $\ln(E_{ij}) = g(Z_{ij})$

- Solve for R, g by minimizing:

$$\sum_{i=1}^{N} \sum_{j=1}^{P} w(Z_{ij}) \left[\ln R_i + \ln \Delta t_j - g(Z_{ij}) \right]^2 + \lambda \sum_{z=Z_{min}}^{Z_{max}} w(z) g''(z)^2$$

The objective

Solve for radiance R and mapping g for each of 256 pixel values to minimize:

$$\sum_{i=1}^{N} \sum_{j=1}^{P} w(Z_{ij}) \left[\ln R_i + \ln \Delta t_j - g(Z_{ij}) \right]^2 + \lambda \sum_{z=Z_{min}}^{Z_{max}} w(z) g''(z)^2$$

give pixels near 0 or 255 less weight

known shutter time for image j

exposure should smoothly increase as pixel intensity increases

radiance at particular pixel site is the same for each image

exposure, as a function of pixel value

The Math

- Let g(z) be the discrete inverse response function
- For each pixel site i in each image j, want:

$$\ln Radiance_i + \ln \Delta t_j = g(Z_{ij})$$

- Solve the overdetermined linear system:

$$\underbrace{\sum_{i=1}^{N}\sum_{j=1}^{P}\left[\ln Radiance_i + \ln \Delta t_j - g(Z_{ij})\right]^2}_{\text{fitting term}} + \underbrace{\lambda \sum_{z=Z_{min}}^{Z_{max}} g''(z)^2}_{\text{smoothness term}}$$

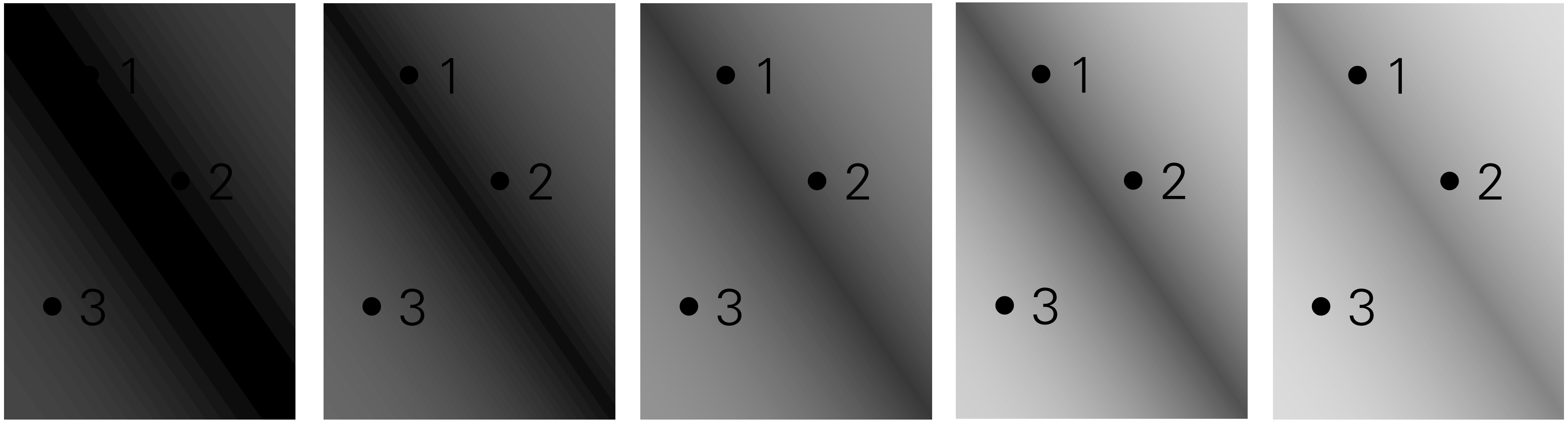
Matlab Code

- 21 lines of code!

```
%
% gsolve.m - Solve for imaging system response function
%
% Given a set of pixel values observed for several pixels in several
% images with different exposure times, this function returns the
% imaging system's response function g as well as the log film irradiance
% values for the observed pixels.
%
% Assumes:
%
%  Zmin = 0
%  Zmax = 255
%
% Arguments:
%
%  Z(i,j) is the pixel values of pixel location number i in image j
%  B(j)   is the log delta t, or log shutter speed, for image j
%  l      is lamdba, the constant that determines the amount of smoothness
%  w(z)   is the weighting function value for pixel value z
%
% Returns:
%
%  g(z)   is the log exposure corresponding to pixel value z
%  lE(i)  is the log film irradiance at pixel location i
%

function [g,lE]=gsolve(Z,B,l,w)

n = 256;

A = zeros(size(Z,1)*size(Z,2)+n+1,n+size(Z,1));
b = zeros(size(A,1),1);

%% Include the data-fitting equations

k = 1;
for i=1:size(Z,1)
  for j=1:size(Z,2)
    wij = w(Z(i,j)+1);
    A(k,Z(i,j)+1) = wij;  A(k,n+i) = -wij;       b(k,1) = wij * B(i,j);
    k=k+1;
  end
end

%% Fix the curve by setting its middle value to 0

A(k,129) = 1;
k=k+1;

%% Include the smoothness equations

for i=1:n-2
  A(k,i)=l*w(i+1);        A(k,i+1)=-2*l*w(i+1);  A(k,i+2)=l*w(i+1);
  k=k+1;
end

%% Solve the system using SVD

x = A\b;

g = x(1:n);
lE = x(n+1:size(x,1));
```

Illustration

Exposure stack



$\Delta t =$ 1/64 sec | $\Delta t =$ 1/16 sec | $\Delta t =$ 1/4 sec | $\Delta t =$ 1 sec | $\Delta t =$ 4 sec

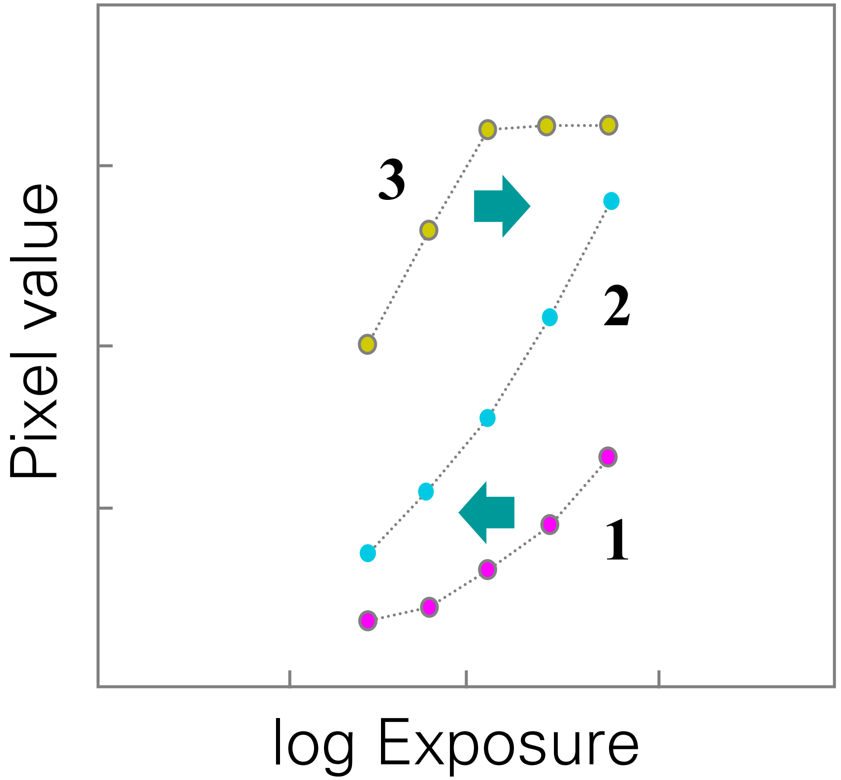
Pixel Value Z = f(Exposure)

Exposure = Radiance * Dt

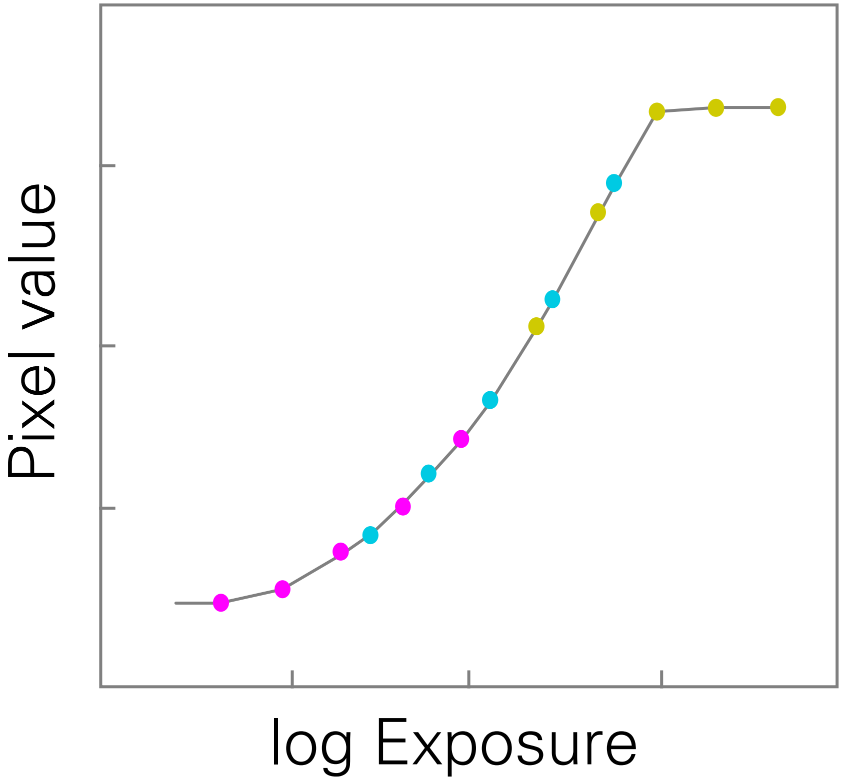
log Exposure = log Radiance + log Dt

Response Curve

Assuming unit radiance
for each pixel



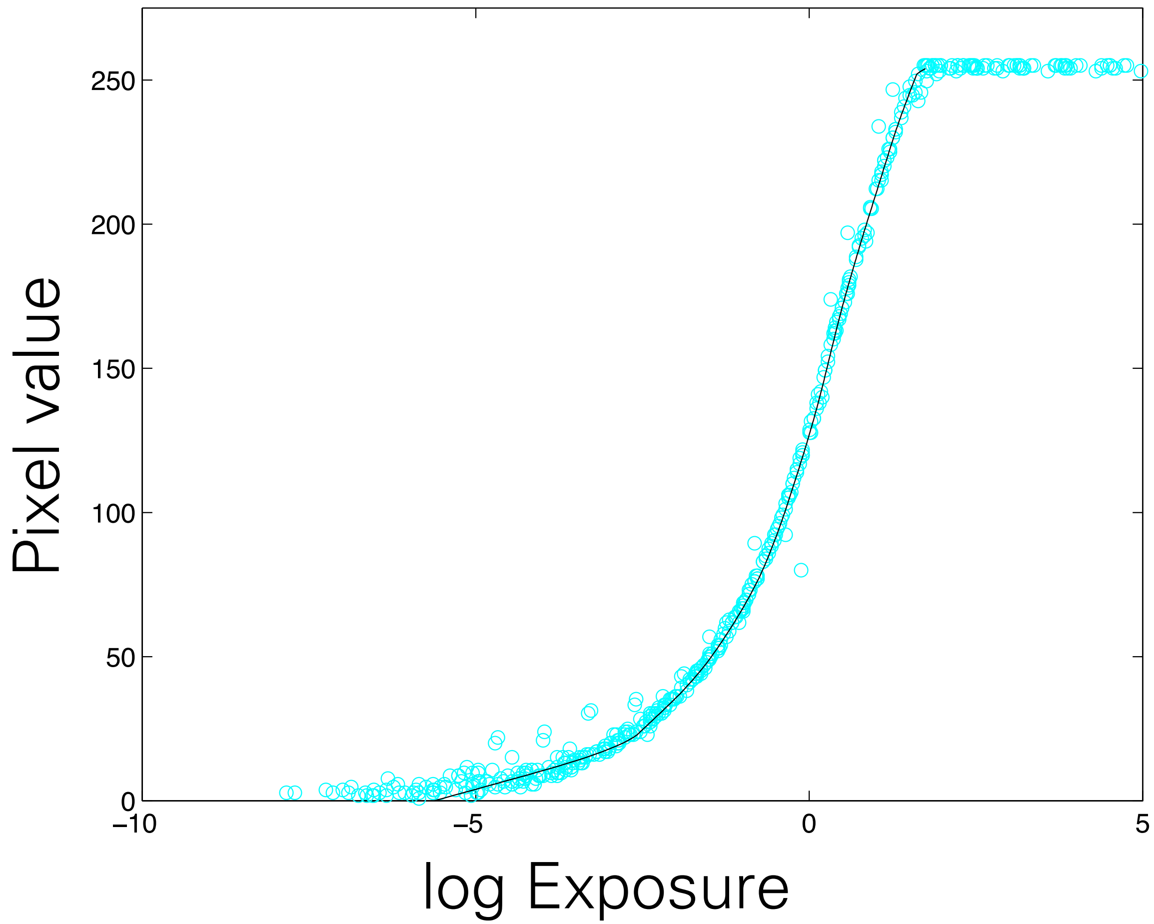
After adjusting radiances to obtain
a smooth response curve



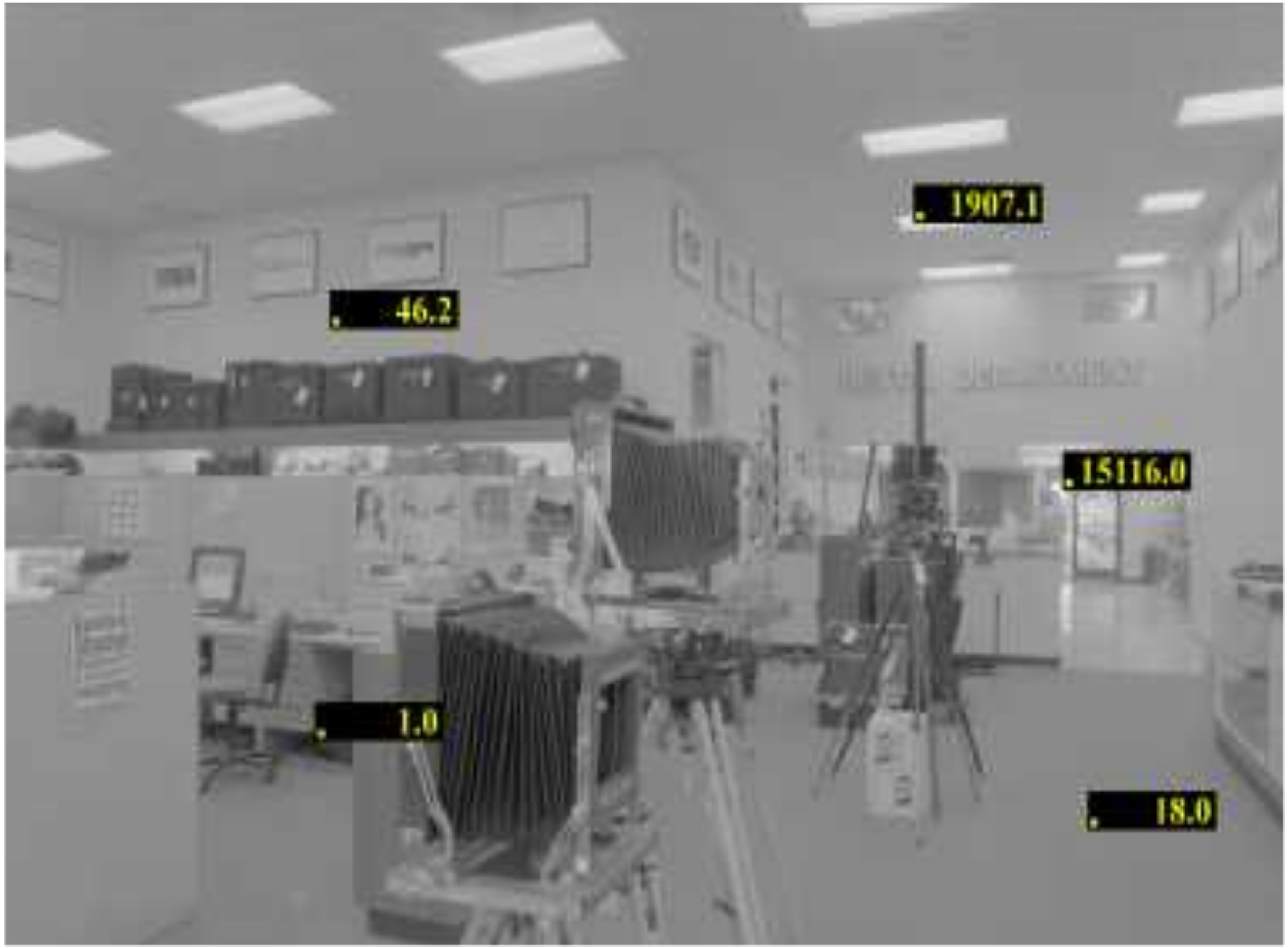
Response Curve

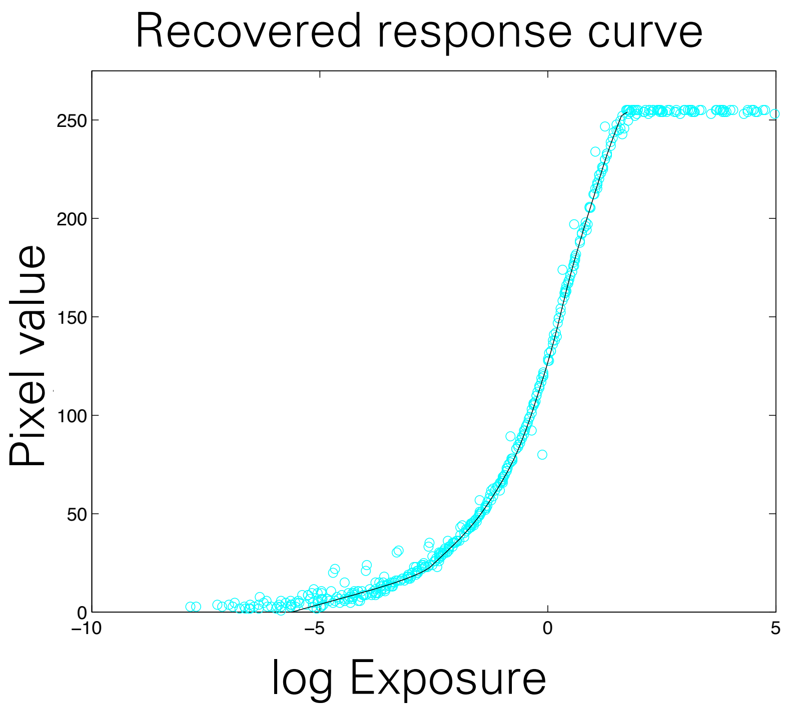
Kodak DCS460
1/30 to 30 sec

Recovered response curve

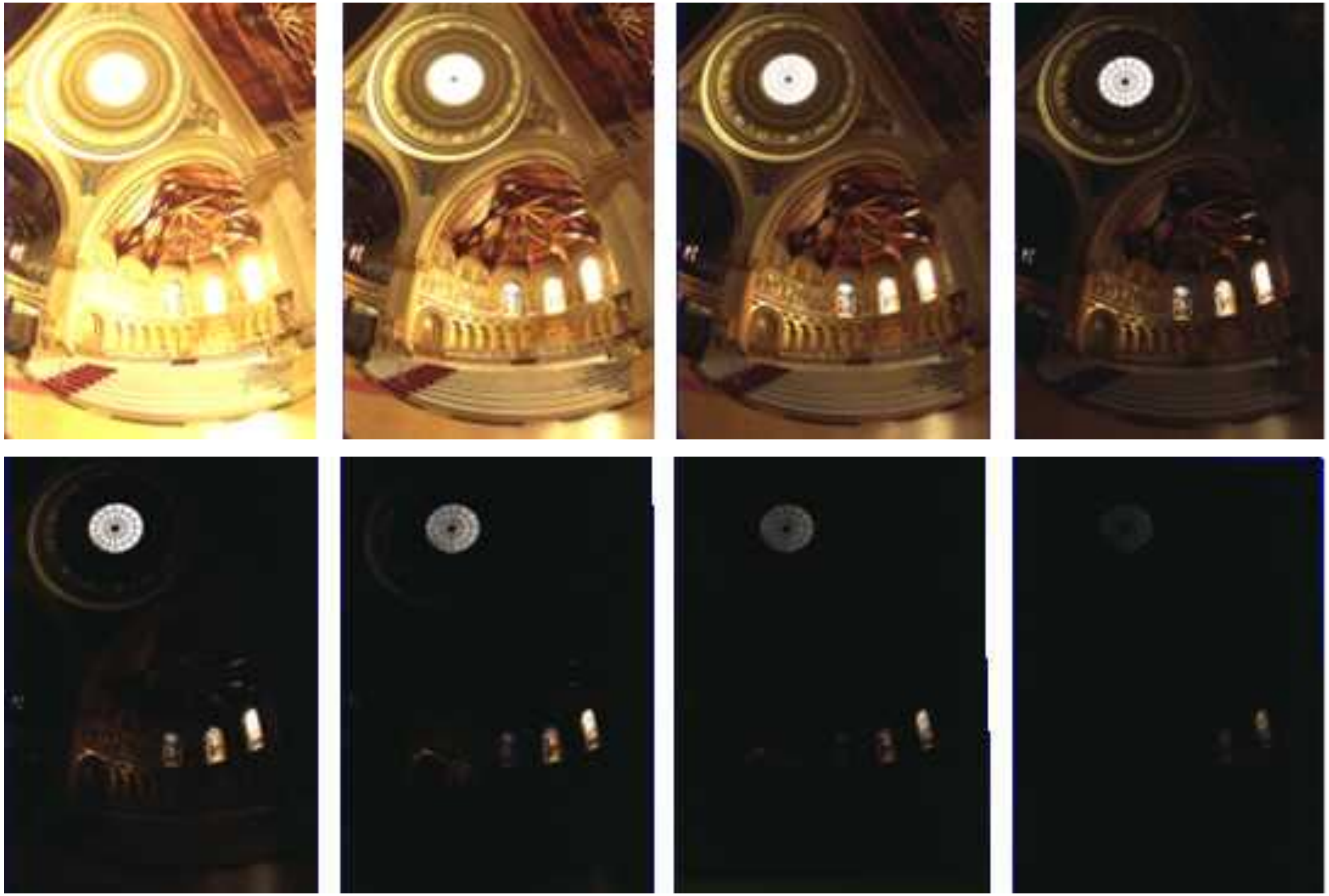


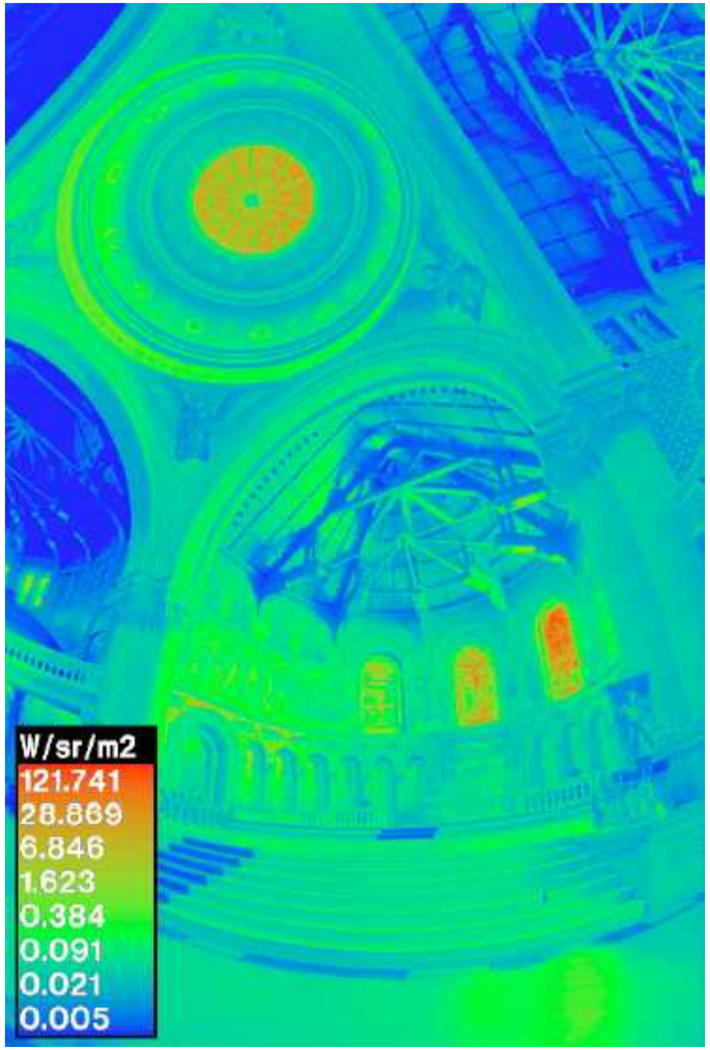
Radiance Image





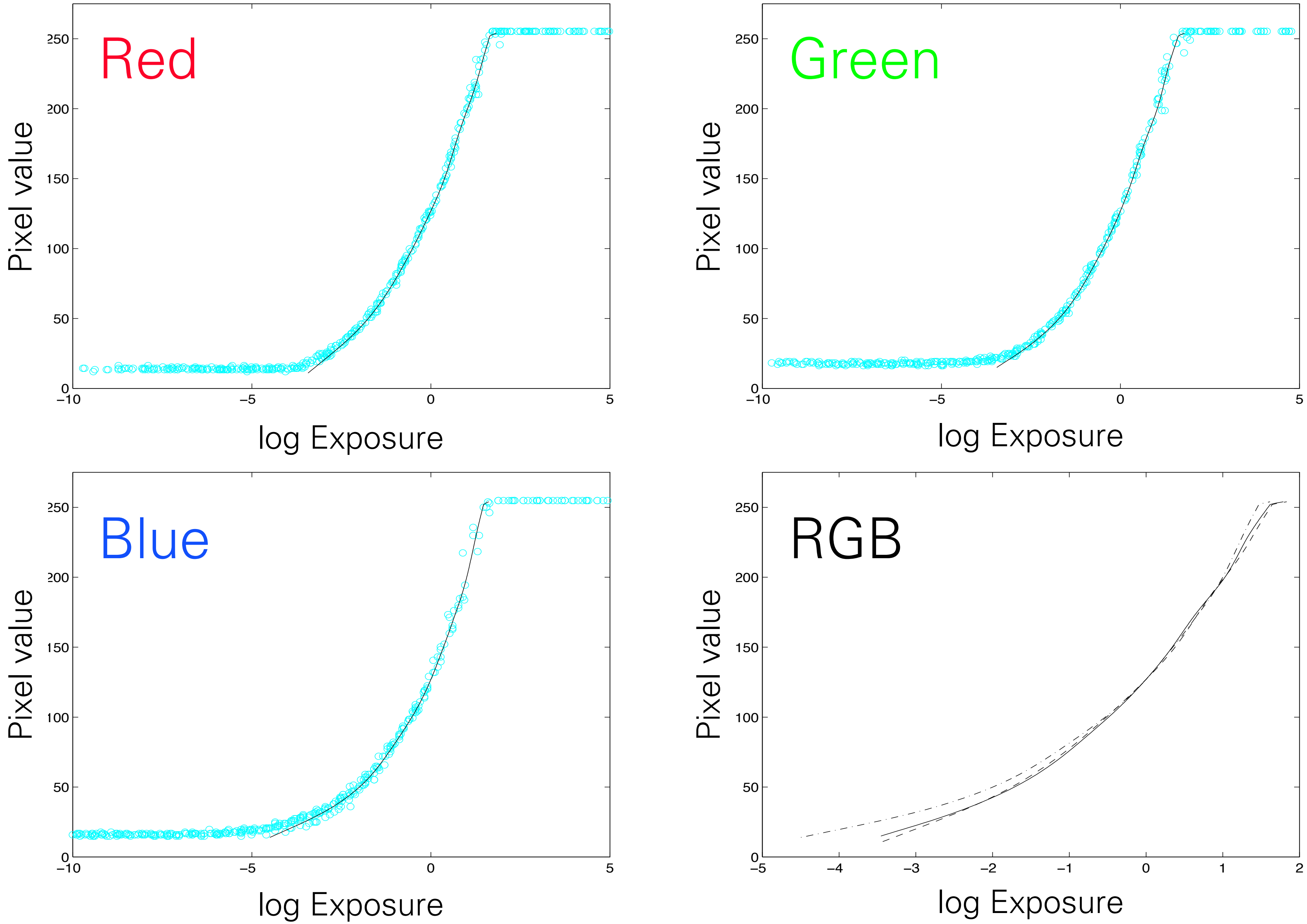
Radiance Image





Kodak Gold ASA 100, PhotoCD

Recovered Response Curves

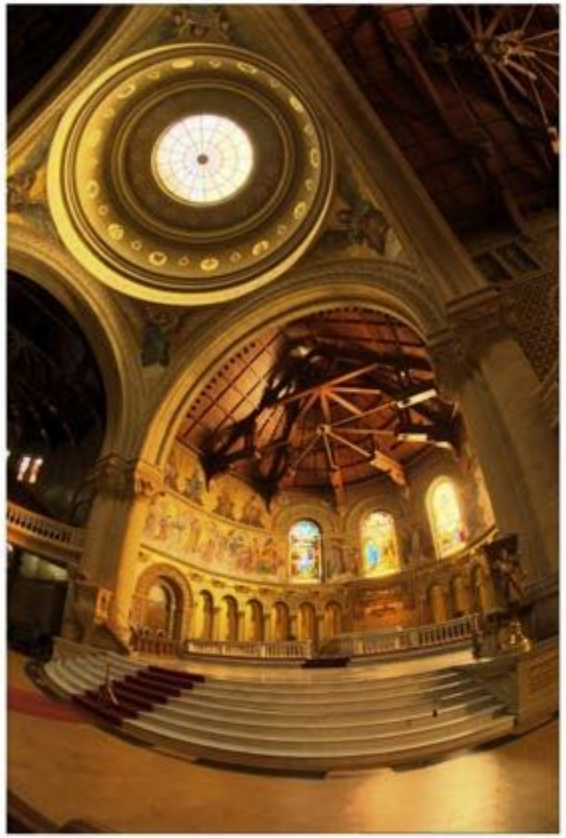


Other aspects of HDR imaging

Relative vs absolute flux

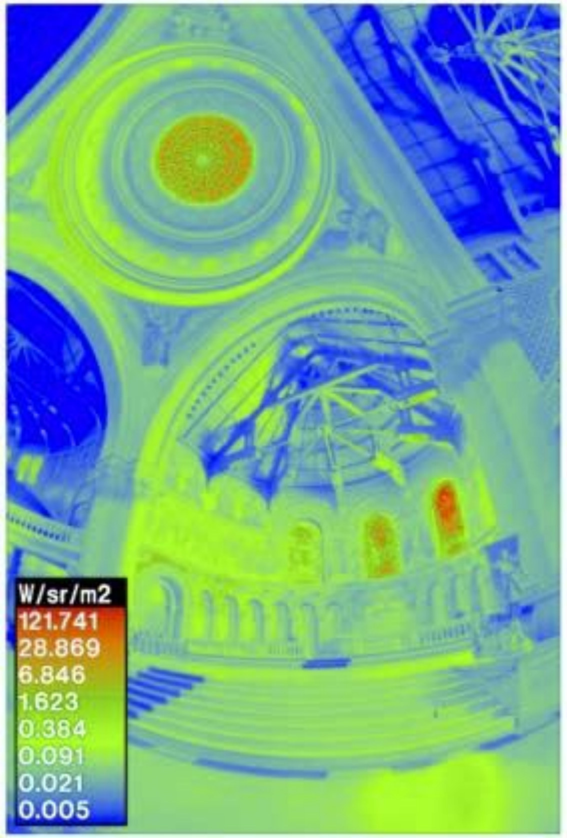
Final fused HDR image gives flux only up to a global scale

- If we know exact flux at one point, we can convert relative HDR image to absolute flux map



HDR image (relative flux)

spotmeter (absolute flux at one point)



absolute flux map

Basic HDR approach

1. Capture multiple LDR images at different exposures
2. Merge them into a single HDR image

Any problems with this approach?

Basic HDR approach

1. Capture multiple LDR images at different exposures

2. Merge them into a single HDR image

Problem: Very sensitive to movement

- Scene must be completely static

- Camera must not move

Most modern automatic HDR solutions include an alignment step before merging exposures

HDR Deghosting

- Family algorithms suggested for eliminating the artefacts occur due to moving objects/camera and/or dynamic backgrounds during HDR reconstruction.
- Mostly the motion is compensated by selecting or removing moving objects and finding alignments between images.

DOI: 10.1111/cgf.12593
EUROGRAPHICS 2015/ K. Hormann and O. Staadt
(Guest Editors)

Volume 34 (2015), Number 2
STAR – State of The Art Report

The State of the Art in HDR Deghosting: A Survey and Evaluation

Okan Tarhan Tursun[1], Ahmet Oğuz Akyüz[1], Aykut Erdem[2] and Erkut Erdem[2]

[1]Dept. of Computer Engineering, Middle East Technical University, Turkey
[2]Dept. of Computer Engineering, Hacettepe University, Turkey

Abstract

Obtaining a high quality high dynamic range (HDR) image in the presence of camera and object movement has been a long-standing challenge. Many methods, known as HDR deghosting algorithms, have been developed over the past ten years to undertake this challenge. Each of these algorithms approaches the deghosting problem from a different perspective, providing solutions with different degrees of complexity, solutions that range from rudimentary heuristics to advanced computer vision techniques. The proposed solutions generally differ in two ways: (1) how to detect ghost regions and (2) what to do to eliminate ghosts. Some algorithms choose to completely discard moving objects giving rise to HDR images which only contain the static regions. Some other algorithms try to find the best image to use for each dynamic region. Yet others try to register moving objects from different images in the spirit of maximizing dynamic range in dynamic regions. Furthermore, each algorithm may introduce different types of artifacts as they aim to eliminate ghosts. These artifacts may come in the form of noise, broken objects, under- and over-exposed regions, and residual ghosting. Given the high volume of studies conducted in this field over the recent years, a comprehensive survey of the state of the art is required. Thus, the first goal of this paper is to provide this survey. Secondly, the large number of algorithms brings about the need to classify them. Thus the second goal of this paper is to propose a taxonomy of deghosting algorithms which can be used to group existing and future algorithms into meaningful classes. Thirdly, the existence of a large number of algorithms brings about the need to evaluate their effectiveness, as each new algorithm claims to outperform its precedents. Therefore, the last goal of this paper is to share the results of a subjective experiment which aims to evaluate various state-of-the-art deghosting algorithms.

Categories and Subject Descriptors (according to ACM CCS): I.4.8 [Image Processing and Computer Vision]: Scene Analysis—Motion

1. Introduction

The real world encompasses a wide range of luminance values that exceeds the capabilities of most image capture devices. However, in general it is desirable to capture, store, process, and display this wide range of luminance values. The field of HDR imaging is primarily developed to address this problem, that is to bridge the gap between what is available in the real-world in terms of light levels and what we can do to represent it using digital equipment [RWPD10].

The first stage of the HDR imaging pipeline is *acquisition*. There have been many studies in HDR image and video acquisition, which can be grouped under three categories. The first category consists of the methods that use specialized hardware to directly capture HDR data. The second category consists of the techniques based on reconstructing an HDR image from a set of low dynamic range (LDR) images of the scene with different exposure settings, techniques that are collectively called as multiple exposure methods. The third category consists of the techniques which aim to expand the dynamic range of a normally LDR image – be it through pseudo-multi-exposure or inverse tone mapping [BADC11].

In general, the techniques in the first and third categories produce inherently ghost-free HDR images as they operate on data captured at a single time instance. The techniques in the second category, however, must deal with moving objects as the image capture process takes a longer time due to necessity of capturing multiple exposures. This is due to the fact that the ensuing HDR image reconstruction process sim-

HDR Deghosting

- Family algorithms suggested for eliminating the artefacts occur due to moving objects/camera and/or dynamic backgrounds during HDR reconstruction.
- Mostly the motion is compensated by selecting or removing moving objects and finding alignments between images.

DOI: 10.1111/cgf.12818
EUROGRAPHICS 2016 / J. Jorge and M. Lin (Guest Editors)

Volume 35 (2016), Number 2

An Objective Deghosting Quality Metric for HDR Images

Okan Tarhan Tursun[1], Ahmet Oğuz Akyüz[1], Aykut Erdem[2] and Erkut Erdem[2]

[1]Dept. of Computer Engineering, Middle East Technical University, Turkey
[2]Dept. of Computer Engineering, Hacettepe University, Turkey

(a) Moving people generate blending (red) and visual difference (blue) artifacts. (b) Over-smoothing gives rise to gradient inconsistency (green) artifacts.

Figure 1: Our metric detects several kinds of HDR deghosting artifacts. In (a), Khan et al.'s [KAR06] output is shown in the bottom-left corner and our metric's result in the bottom-right. The same for (b), except Hu et al.'s [HGPS13] deghosting algorithm is used. Exposure sequences are shown on the top. Cyan color occurs due to both gradient and visual difference metrics producing high output.

Abstract

Reconstructing high dynamic range (HDR) images of a complex scene involving moving objects and dynamic backgrounds is prone to artifacts. A large number of methods have been proposed that attempt to alleviate these artifacts, known as HDR deghosting algorithms. Currently, the quality of these algorithms are judged by subjective evaluations, which are tedious to conduct and get quickly outdated as new algorithms are proposed on a rapid basis. In this paper, we propose an objective metric which aims to simplify this process. Our metric takes a stack of input exposures and the deghosting result and produces a set of artifact maps for different types of artifacts. These artifact maps can be combined to yield a single quality score. We performed a subjective experiment involving 52 subjects and 16 different scenes to validate the agreement of our quality scores with subjective judgements and observed a concordance of almost 80%. Our metric also enables a novel application that we call as hybrid deghosting, in which the output of different deghosting algorithms are combined to obtain a superior deghosting result.

Categories and Subject Descriptors (according to ACM CCS): I.4.8 [Image Processing and Computer Vision]: Scene Analysis—Motion

1. Introduction

Due to its low-cost and availability, the most commonly used HDR image capture method remains to be the multiple exposures technique (MET), which involves combining a set of exposures of a scene into a single HDR image [DM97]. The main requirements of this technique are that the camera and the captured scene remain static throughout the capture process. Otherwise, the lack of correspondence between exposures result in what is known as *ghosting artifacts*. While stabilizing a camera can be achieved by using a tripod, ensuring a static scene is much more difficult as most real-world scenes contain dynamic objects. Many deghosting algorithms have been proposed to address this problem ranging from

© 2016 The Author(s)
Computer Graphics Forum © 2016 The Eurographics Association and John Wiley & Sons Ltd. Published by John Wiley & Sons Ltd.

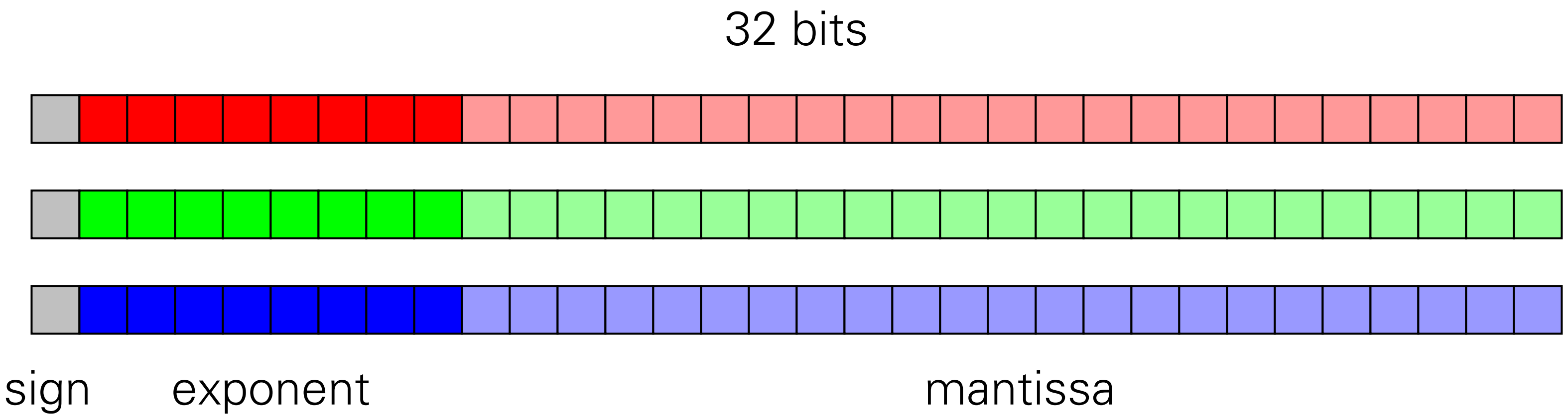
How do we store HDR images?

- Most standard image formats store integer 8-bit images
- Some image formats store integer 12-bit or 16-bit images
- HDR images are floating point 32-bit or 64-bit images

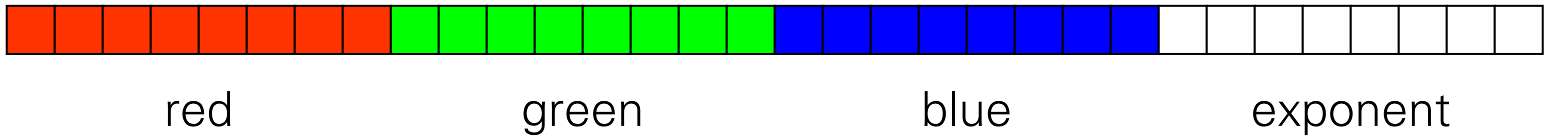
How do we store HDR images?

Use specialized image formats for HDR images

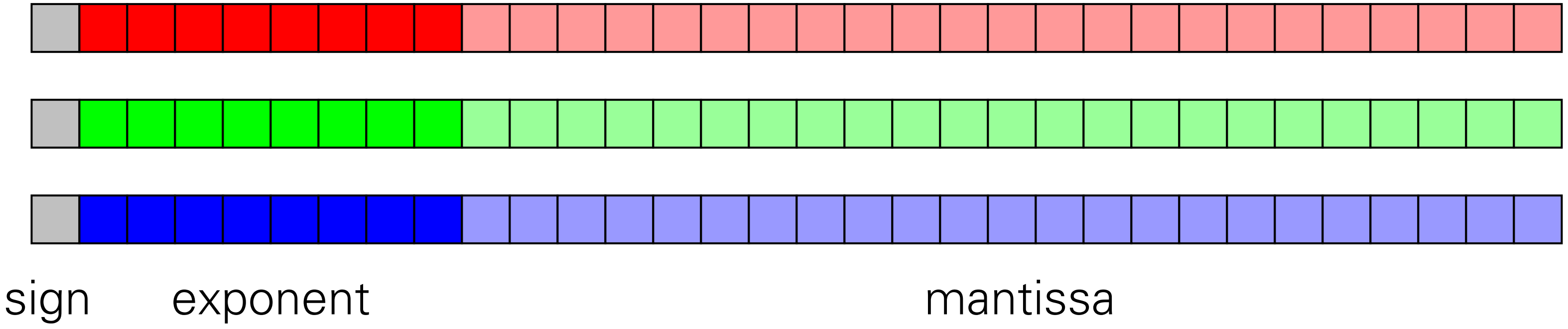
portable float map (.pfm)
- very simple to implement



Radiance format (.hdr)
- supported by Matlab



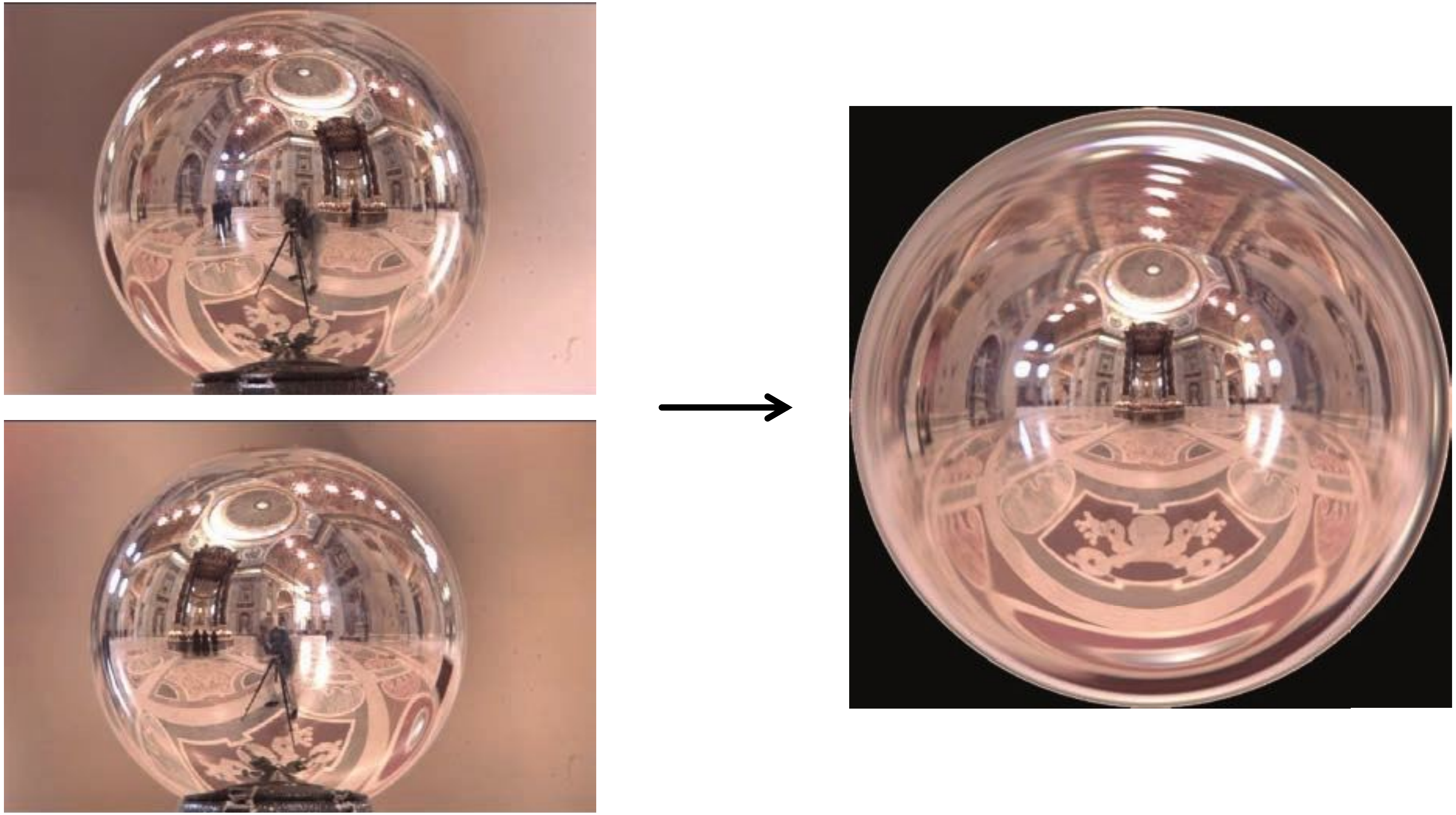
OpenEXR format (.exr)
- multiple extra features



Another type of HDR images

Light probes: place a chrome sphere in the scene and capture an HDR image

- Used to measure real-world illumination environments ("environment maps")

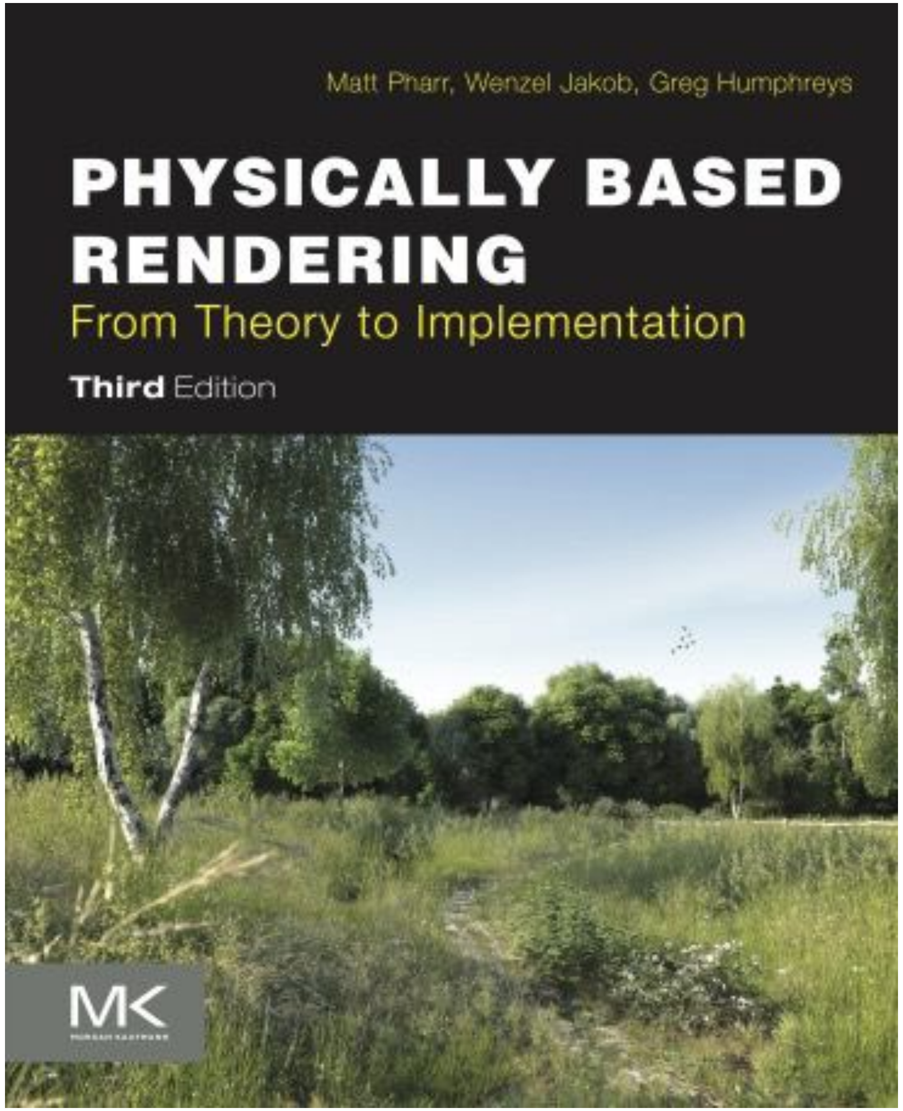


Application: image-based relighting

Another way to create HDR images

Physics-based renderers simulate flux maps (relative or absolute)

- Their outputs are very often HDR images



Our devices do not match the real world

- 10:1 photographic print (higher for glossy paper)
- 20:1 artist's paints
- 200:1 slide film
- 500:1 negative film
- 1000:1 LCD display
- 2000:1 digital SLR (at 12 bits)
- 100000:1 real world

HDR imaging and tonemapping are distinct techniques with different goals

Two challenges:

HDR imaging compensates for sensor limitations

1. HDR imaging – which parts of the world do we measure in the 8-14 bits available to our sensor?
2. Tonemapping – which parts of the world do we show in the 4-10 bits available to our display?

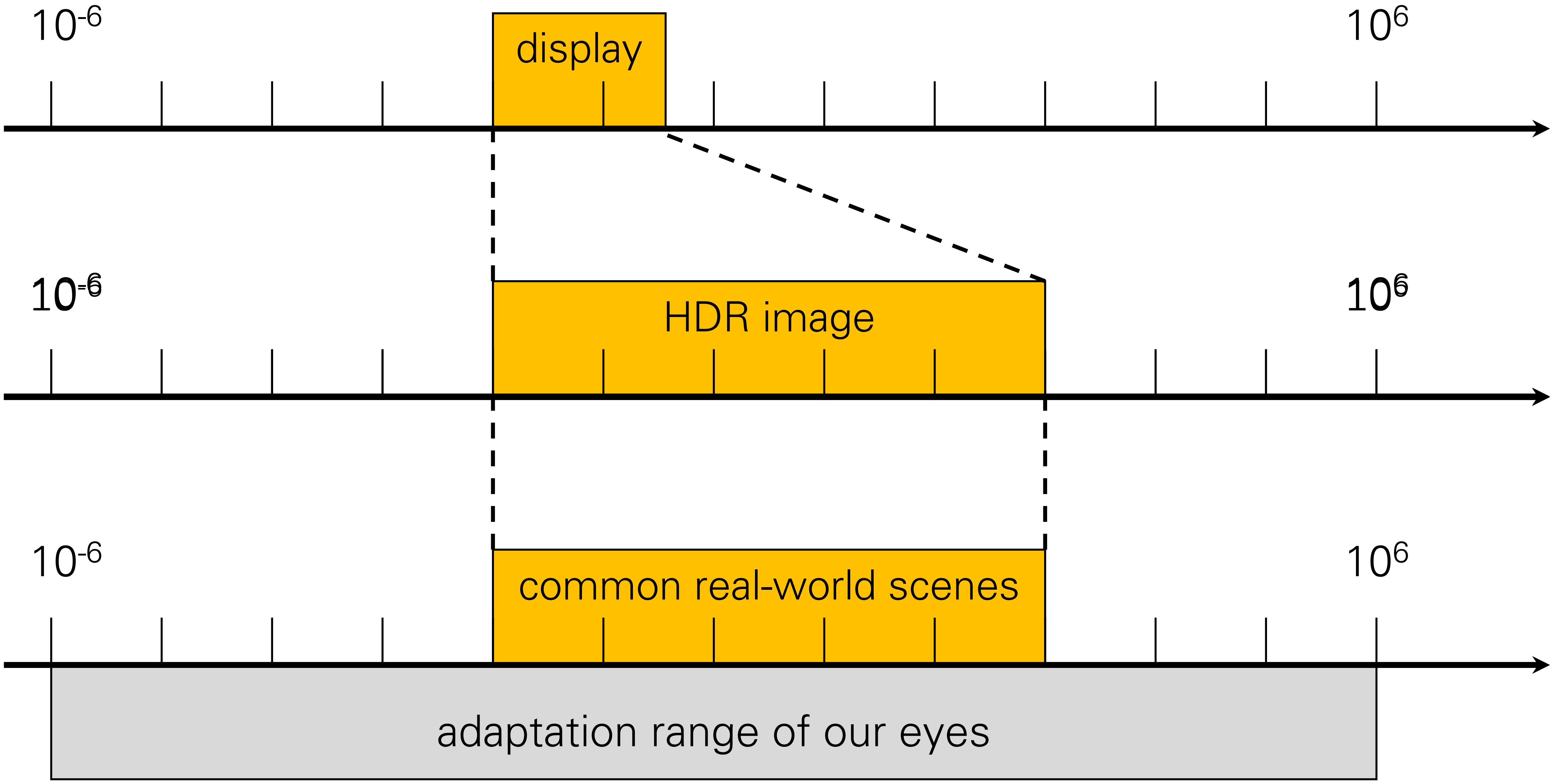
Tonemapping compensates for display limitations

Today's Lecture

- Controlling exposure
- High-dynamic-range imaging
- Tonemapping

Tonemapping

How do we display our HDR images?

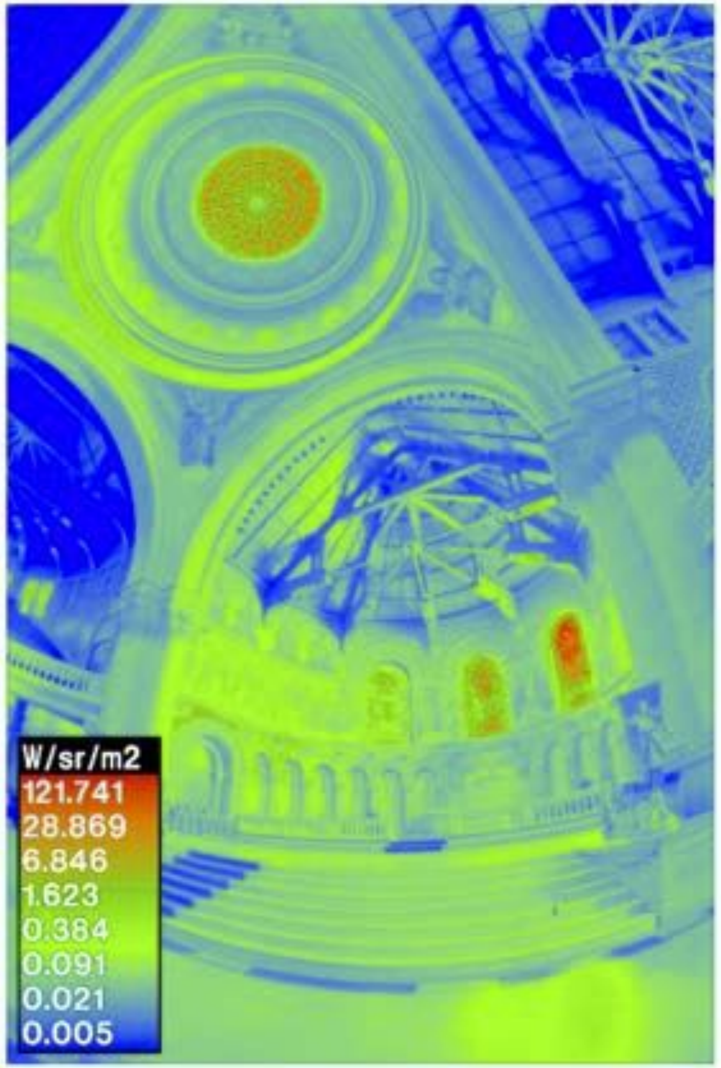


Tonemapping

- Called tone mapping operators
- Two general categories:
  - Global (spatially invariant)
  - Local (spatially varying)

Linear scaling

Scale image so that maximum value equals 1.

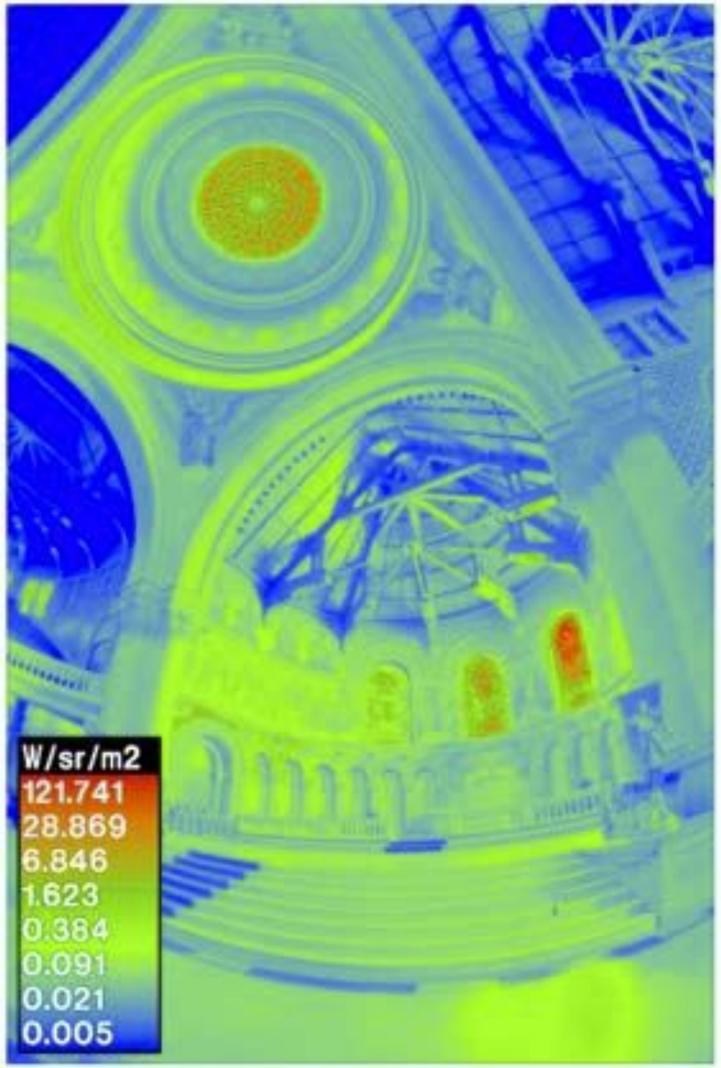


→

HDR image looks underexposed because of the display's limited dynamic range, but is not actually underexposed.

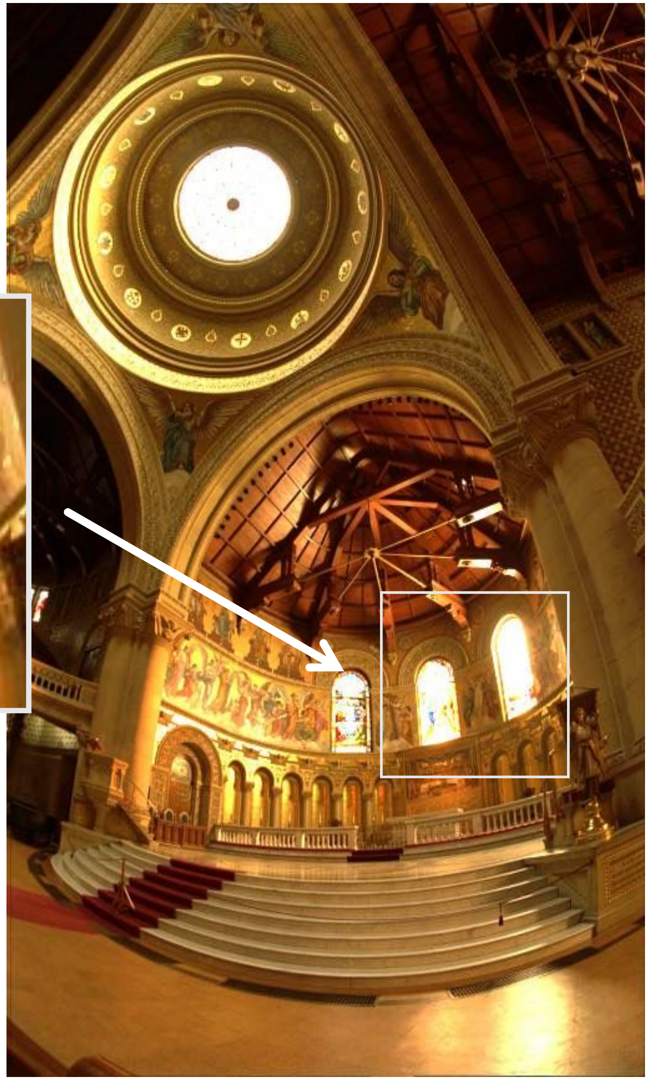
Linear scaling

Scale image so that 10% value equals 1.



→

HDR image looks saturated because of the display's limited dynamic range, but is not actually saturated.

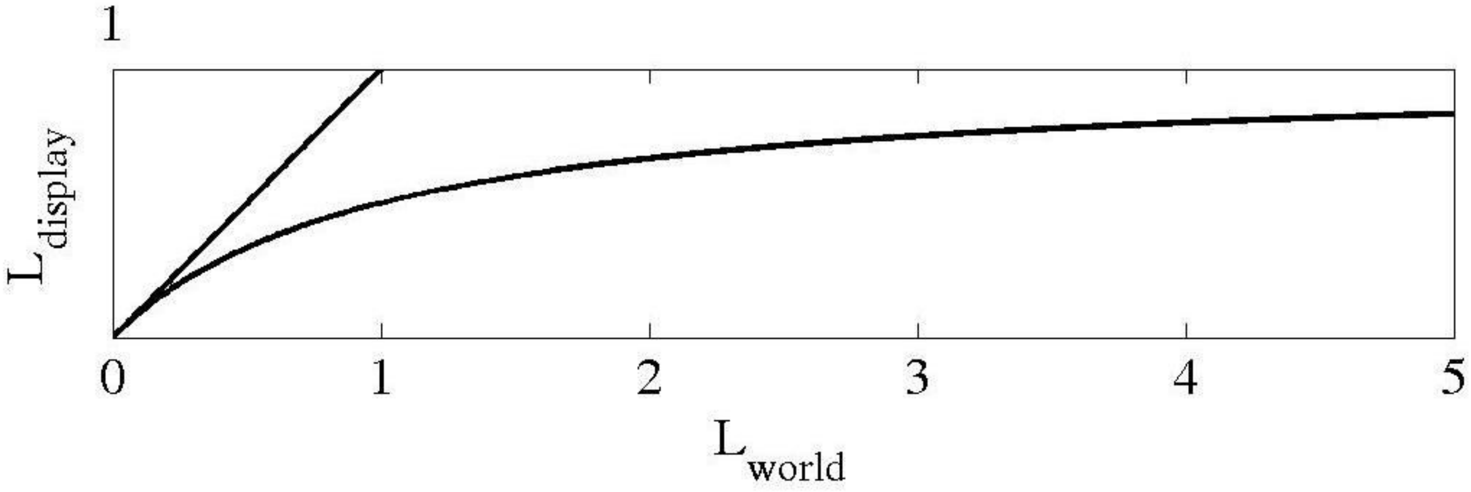


Can you think of something better?

Photographic tonemapping

Apply the same non-linear scaling to all pixels in the image so that:

- Bring everything within range → asymptote to 1
- Leave dark areas alone → slope = 1 near 0



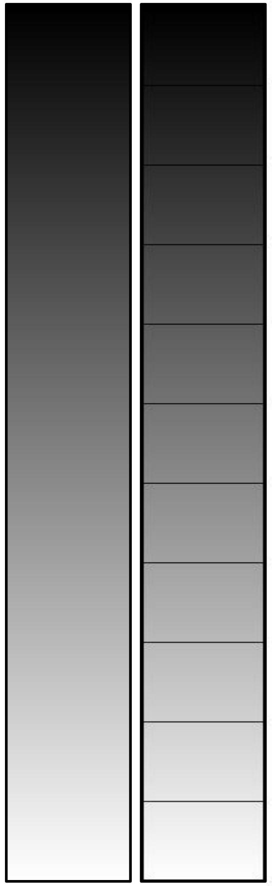
$$I_{display} = \frac{I_{HDR}}{1 + I_{HDR}}$$

(exact formula more complicated)

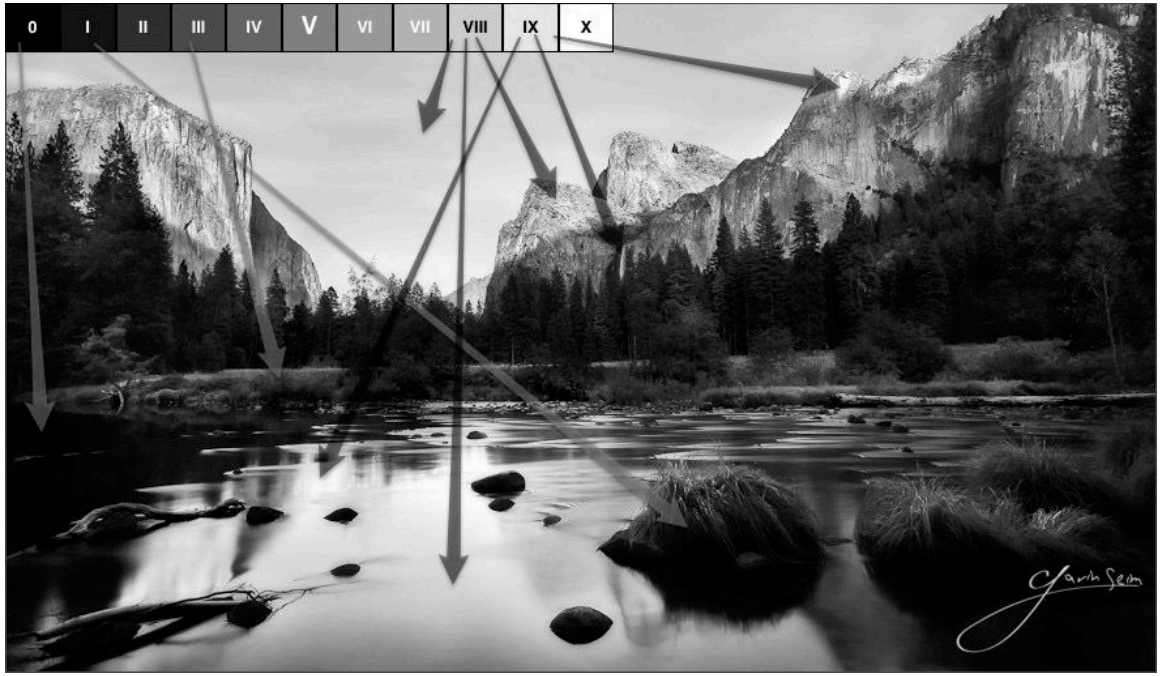
- Photographic because designed to approximate film zone system.
- Perceptually motivated, as it approximates our eye's response curve.

What is the zone system?

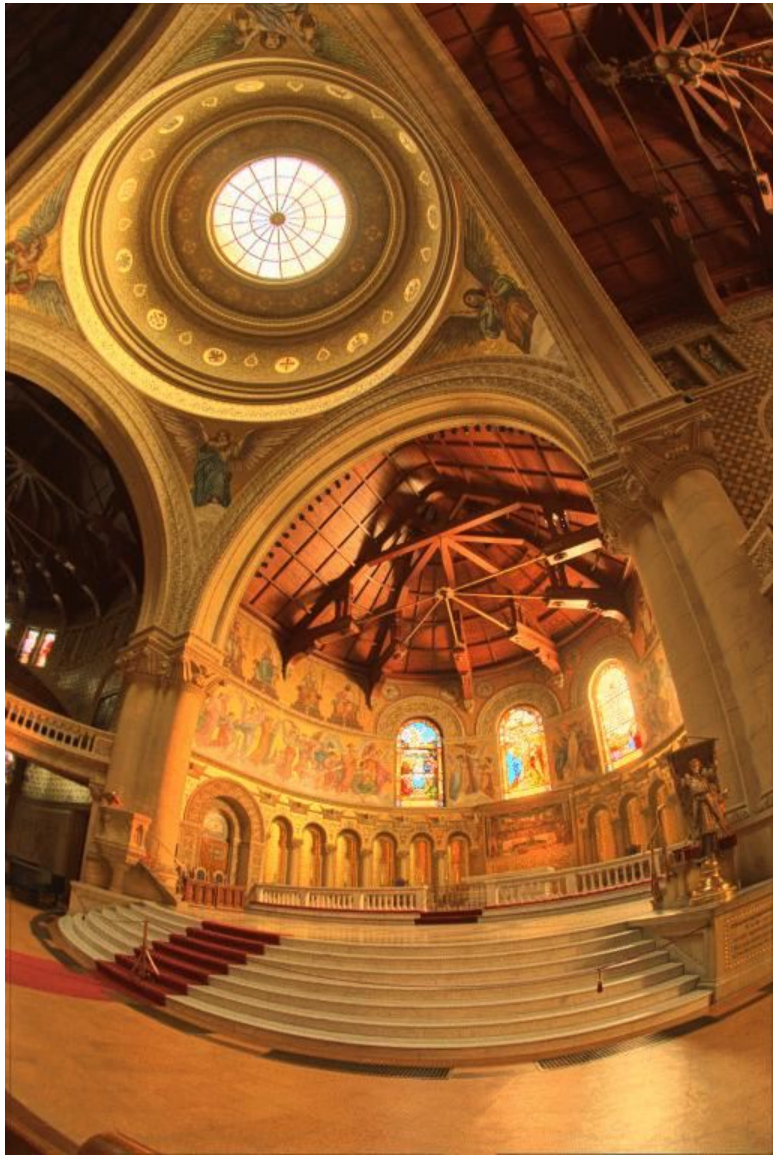
- Technique formulated by Ansel Adams for film development.
- Still used with digital photography.



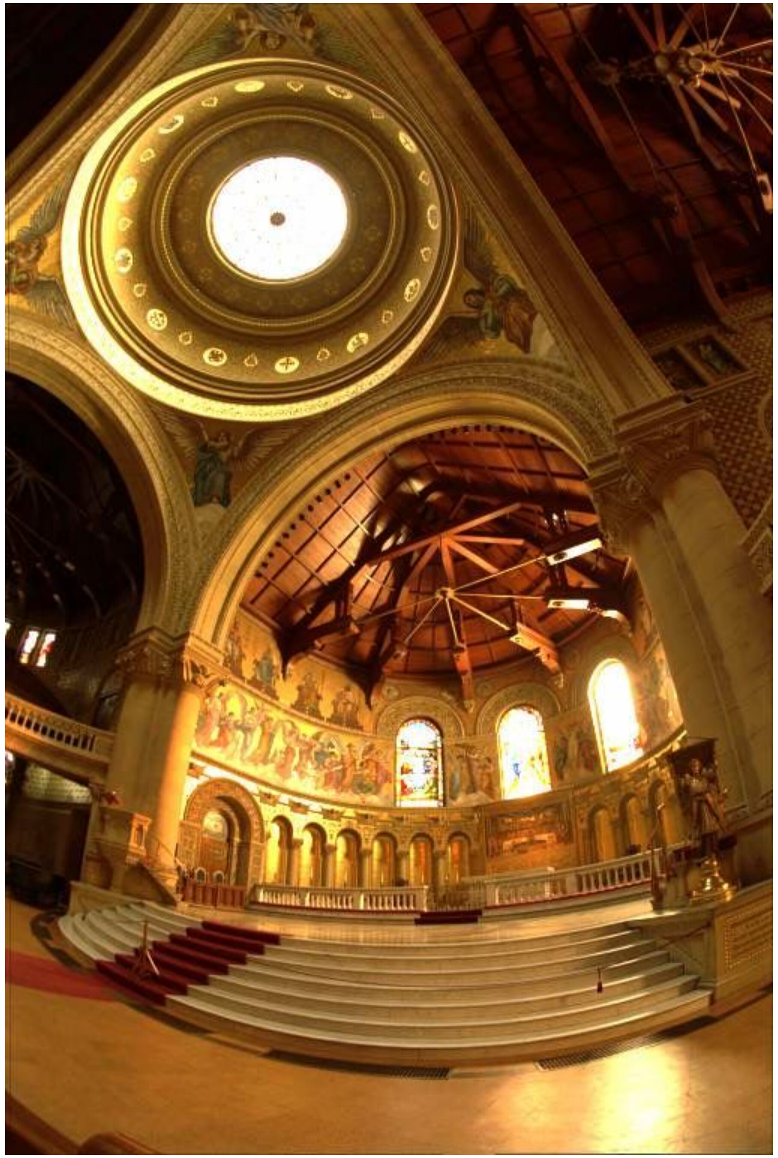
Zone	Description
0	Pure black
I	Near black, with slight tonality but no texture
II	Textured black; the darkest part of the image in which slight detail is recorded
III	Average dark materials and low values showing adequate texture
IV	Average dark foliage, dark stone, or landscape shadows
V	Middle gray: clear north sky; dark skin, average weathered wood
VI	Average Caucasian skin; light stone; shadows on snow in sunlit landscapes
VII	Very light skin; shadows in snow with acute side lighting
VIII	Lightest tone with texture: textured snow
IX	Slight tone without texture; glaring snow
X	Pure white: light sources and specular reflections



Examples



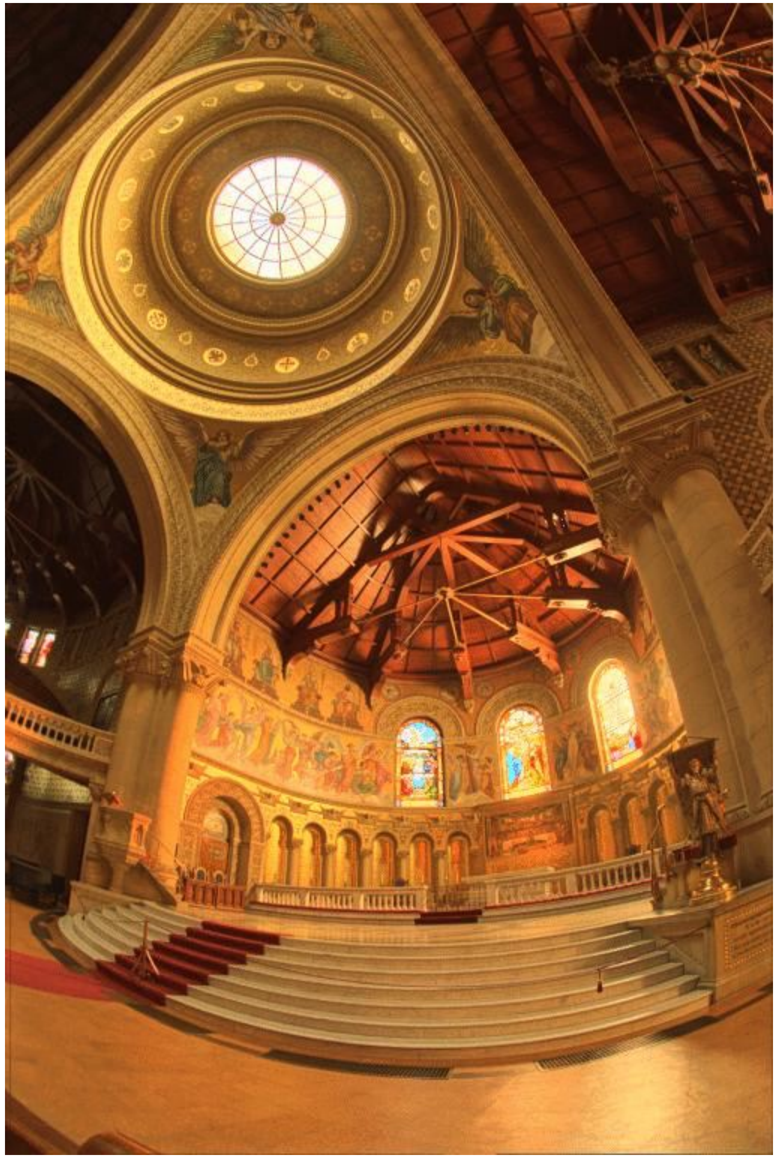
photographic tonemapping



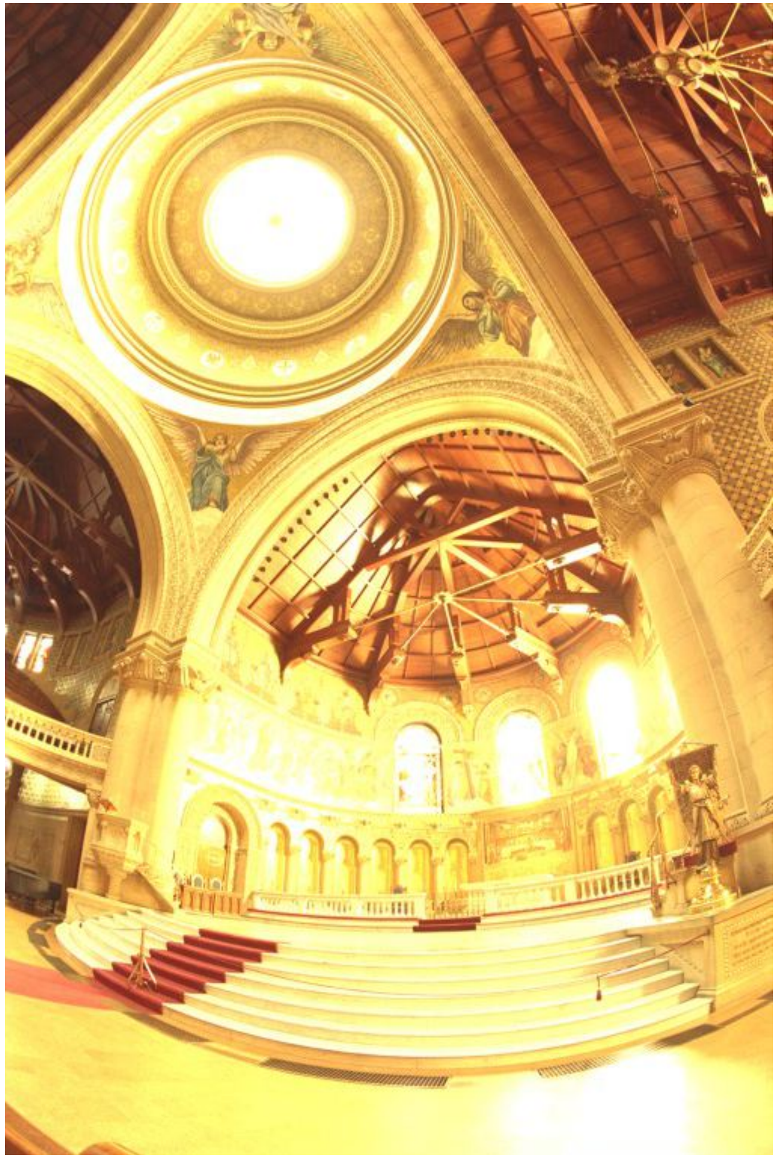
linear scaling (map 10% to 1)

linear scaling (map 100% to 1)

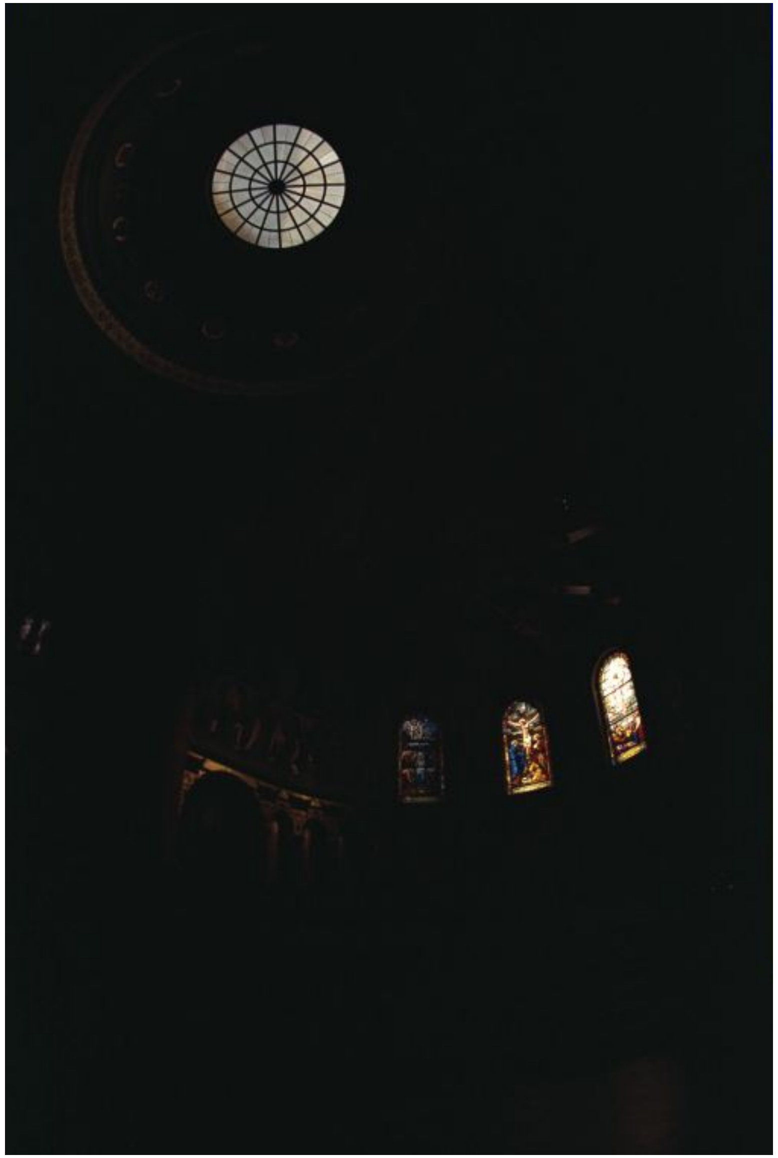
Compare with LDR images



photographic tonemapping



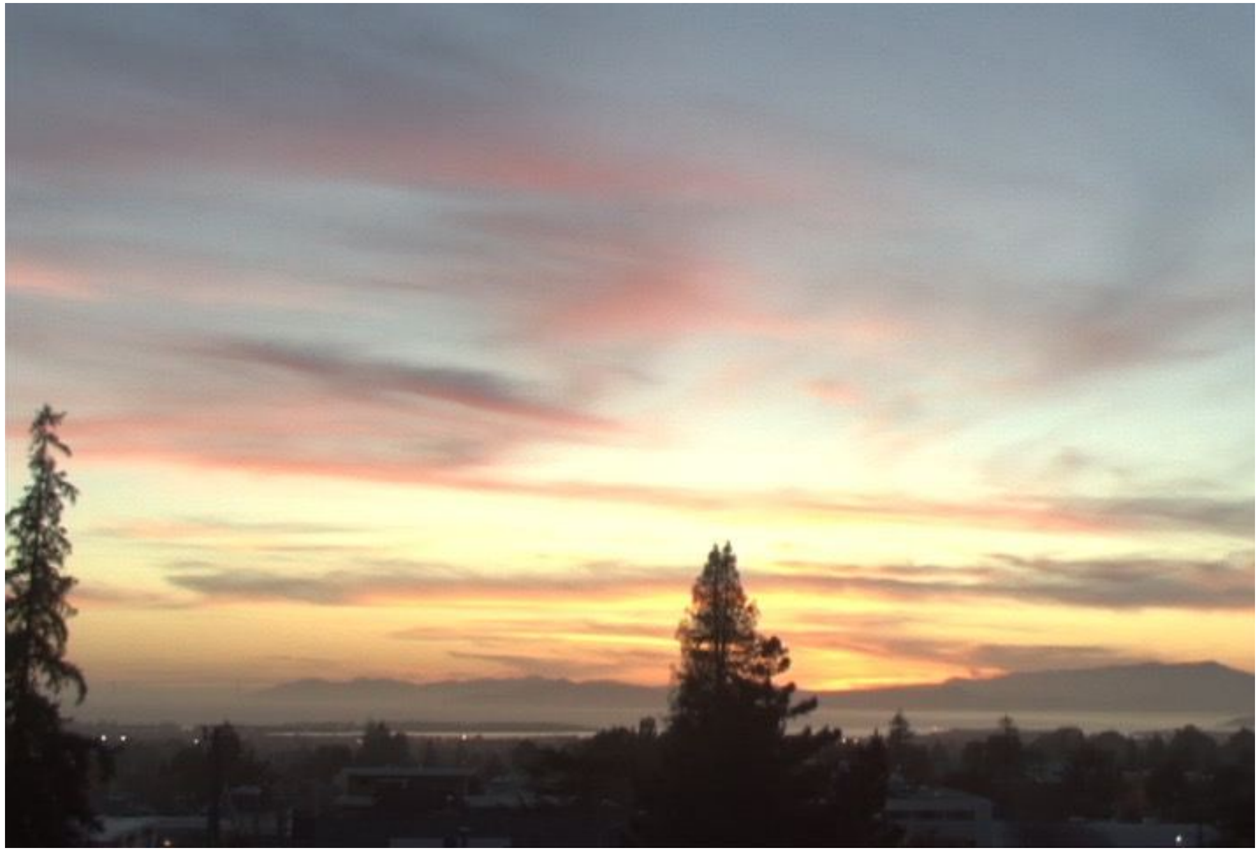
high exposure



low exposure

Dealing with color

If we tonemap all channels the same, colors are washed out

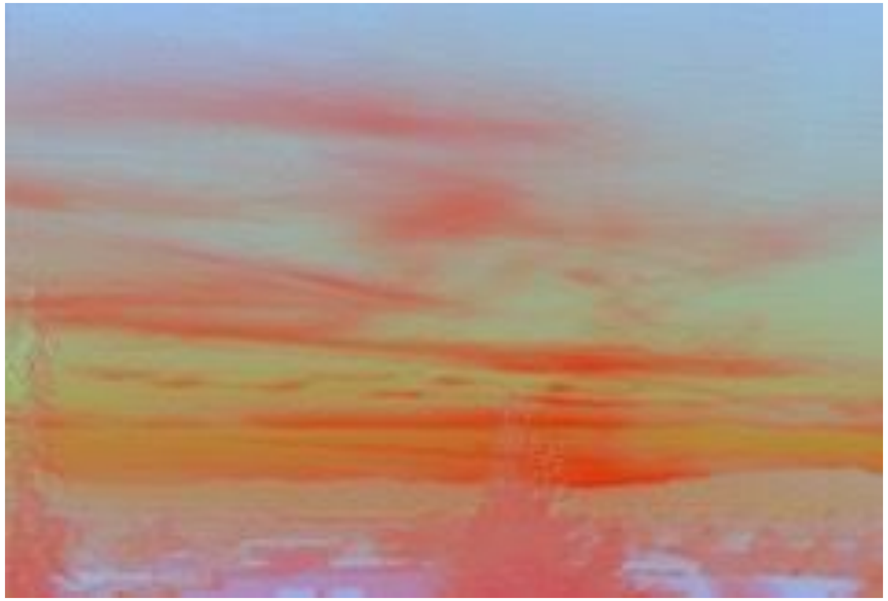


Can you think of a way to deal with this?

Intensity-only tonemapping

tonemap intensity (e.g., luminance Y in xyY)

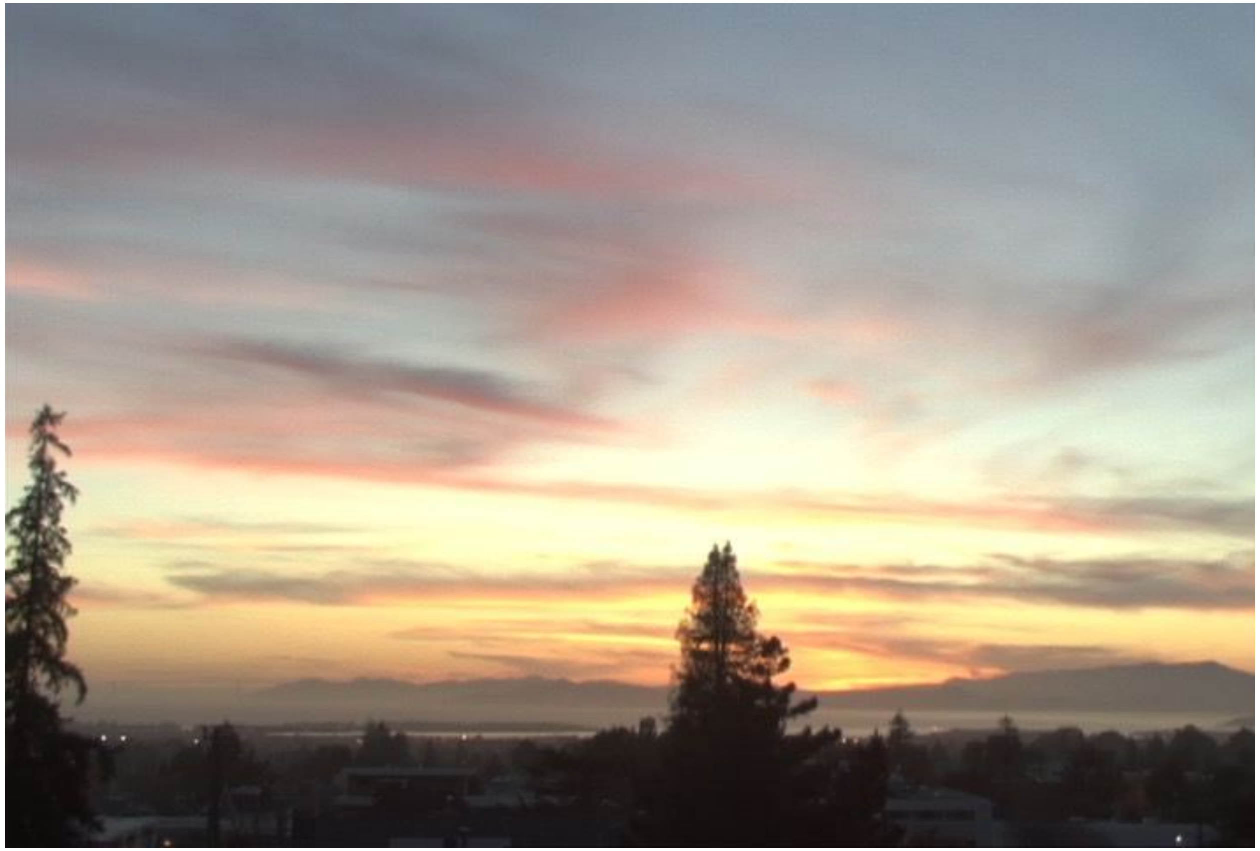
leave color the same (e.g., xy in xyY)



How would you implement this?

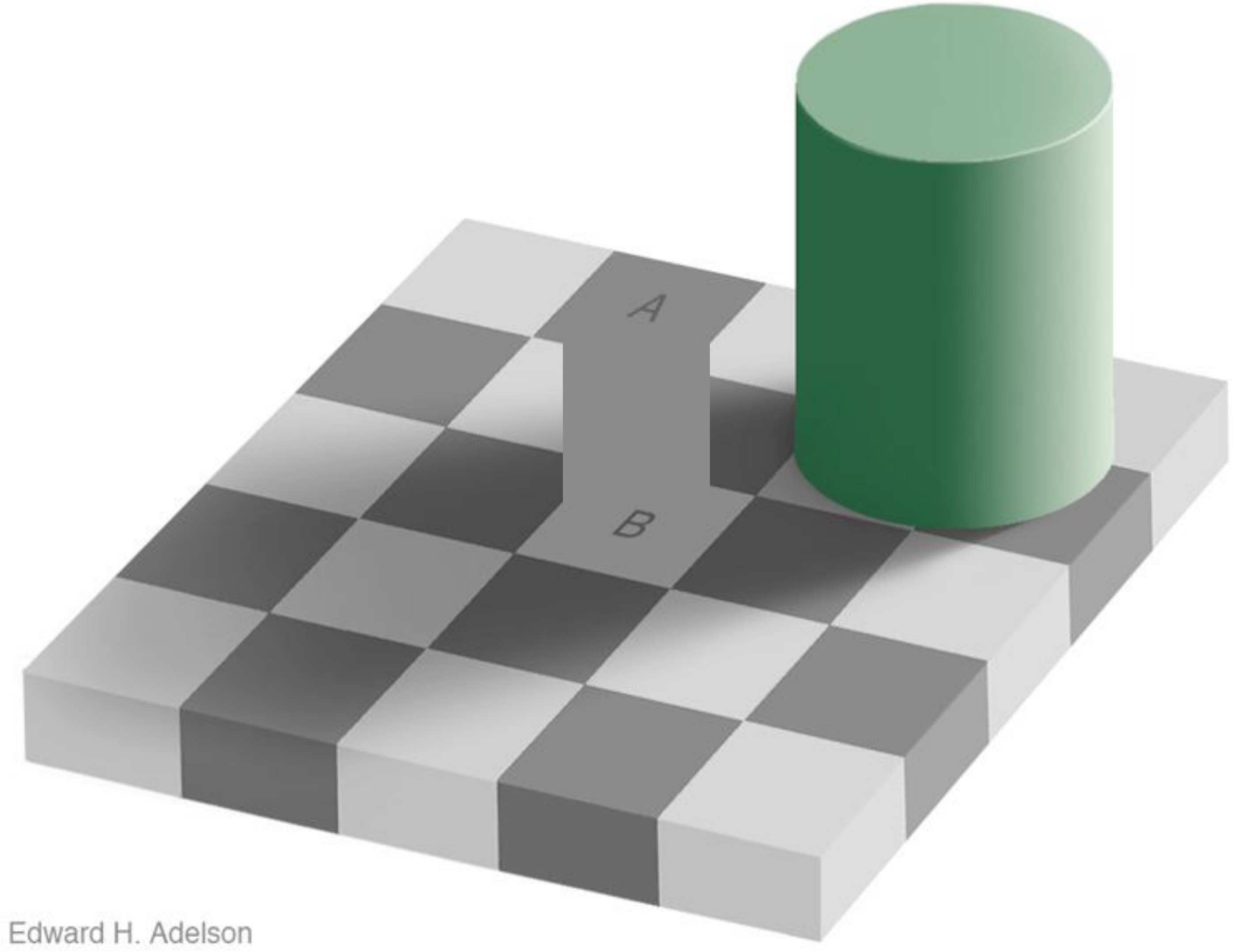
Comparison

Color now OK, but some details are washed out due to loss of contrast



Can you think of a way to deal with this?

The importance of local contrast



Purposes of tone mapping

Technical:

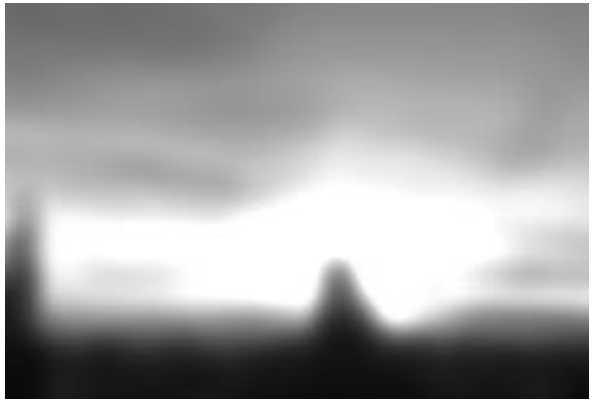
- fitting a wide range of values into a small space while preserving differences between values as much as possible

Artistic

- reproduce what the photographer/artist feels she saw
- stylize the look of a photo

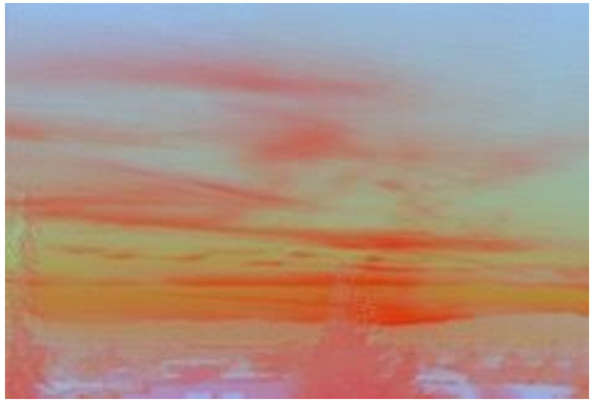
Low-frequency intensity-only tonemapping

tonemap low-frequency intensity component



leave high-frequency intensity component the same

leave color the same



How would you implement this?

Comparison

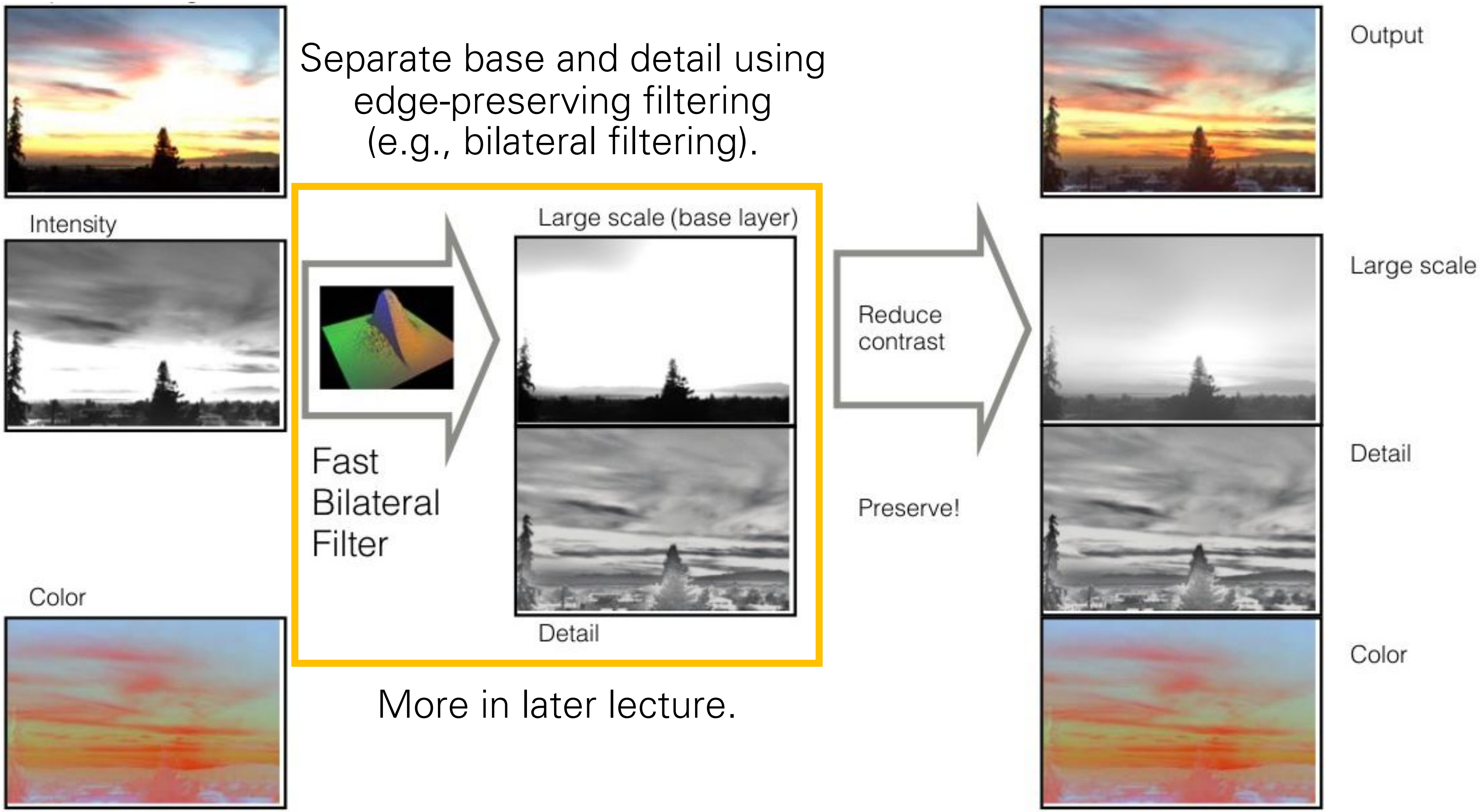
We got nice color and contrast, but now we’ve run into the halo plague

Can you think of a way to deal with this?

Edge-aware filtering and tonemapping

Separate base and detail using edge-preserving filtering (e.g., bilateral filtering).



More in later lecture.

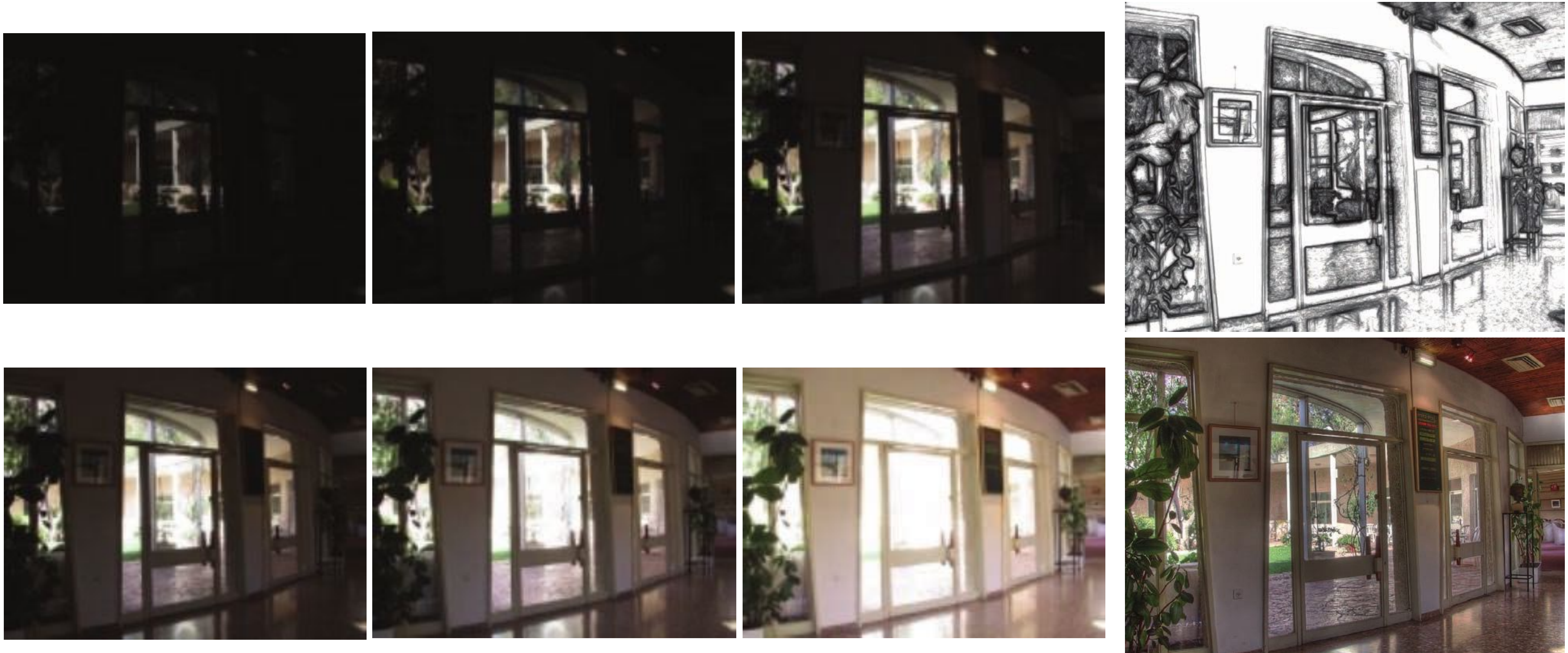
Comparison

We fixed the halos without losing contrast

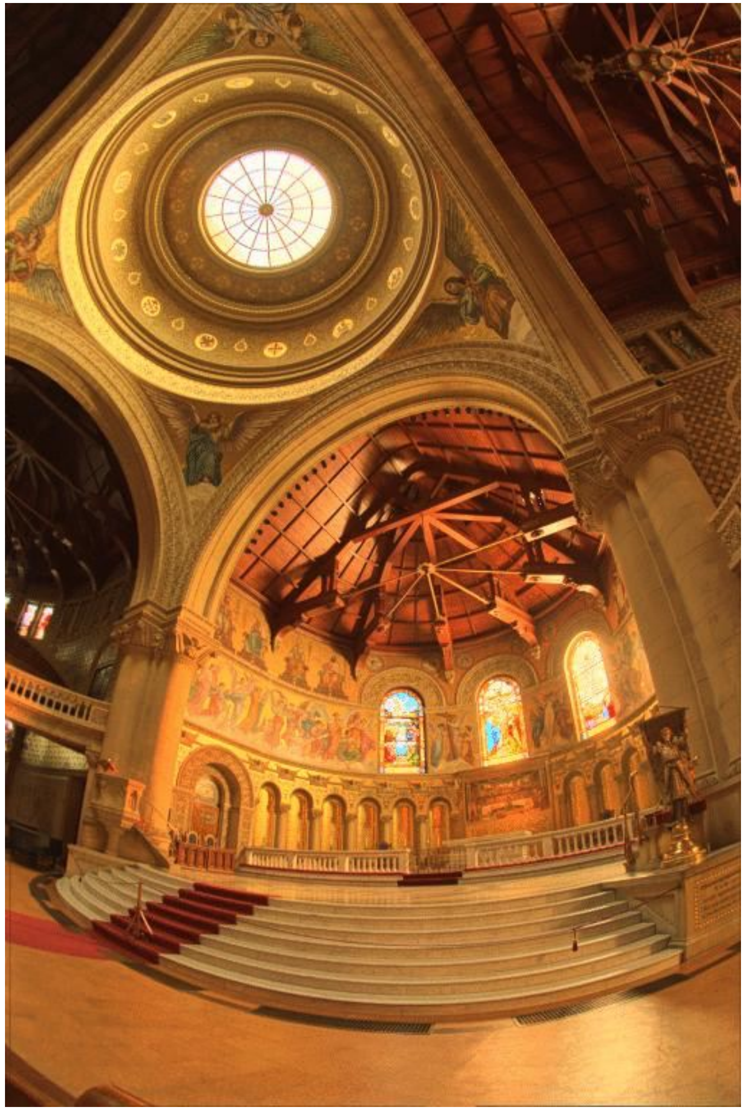
Gradient-domain processing and tonemapping

Compute gradients, scale and merge them, then integrate (solve Poisson problem).

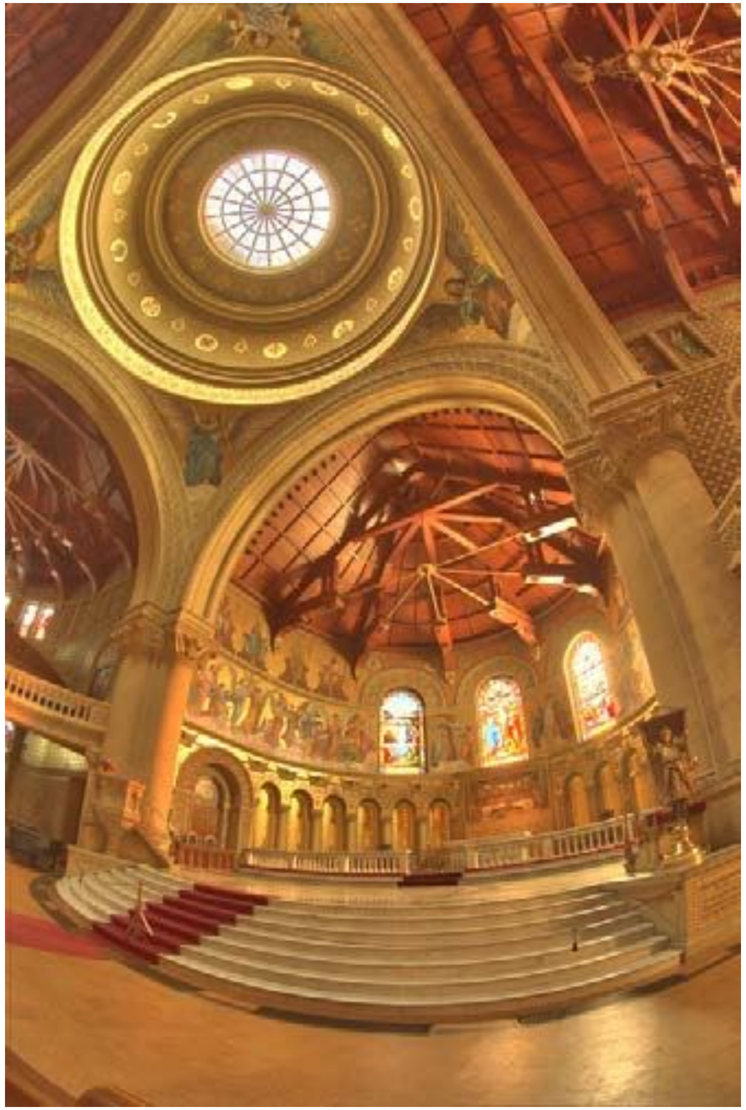
- More in later lecture.



Comparison (which one do you like better?)



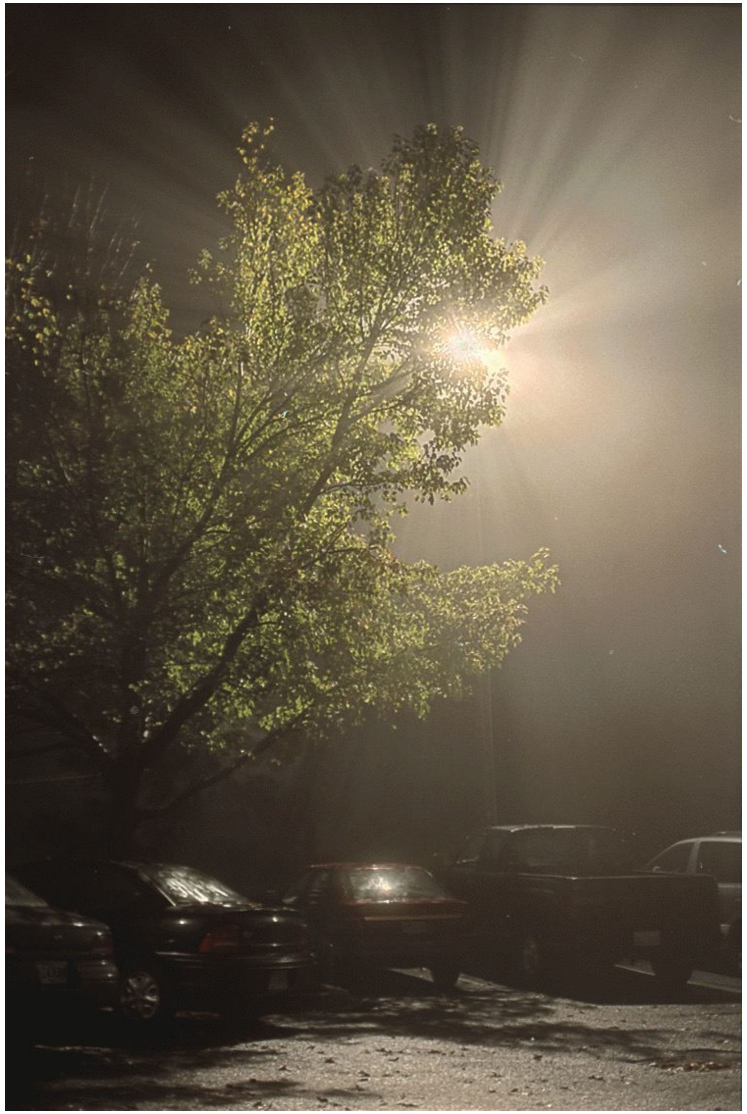
photographic



bilateral filtering

gradient-domain

Comparison (which one do you like better?)

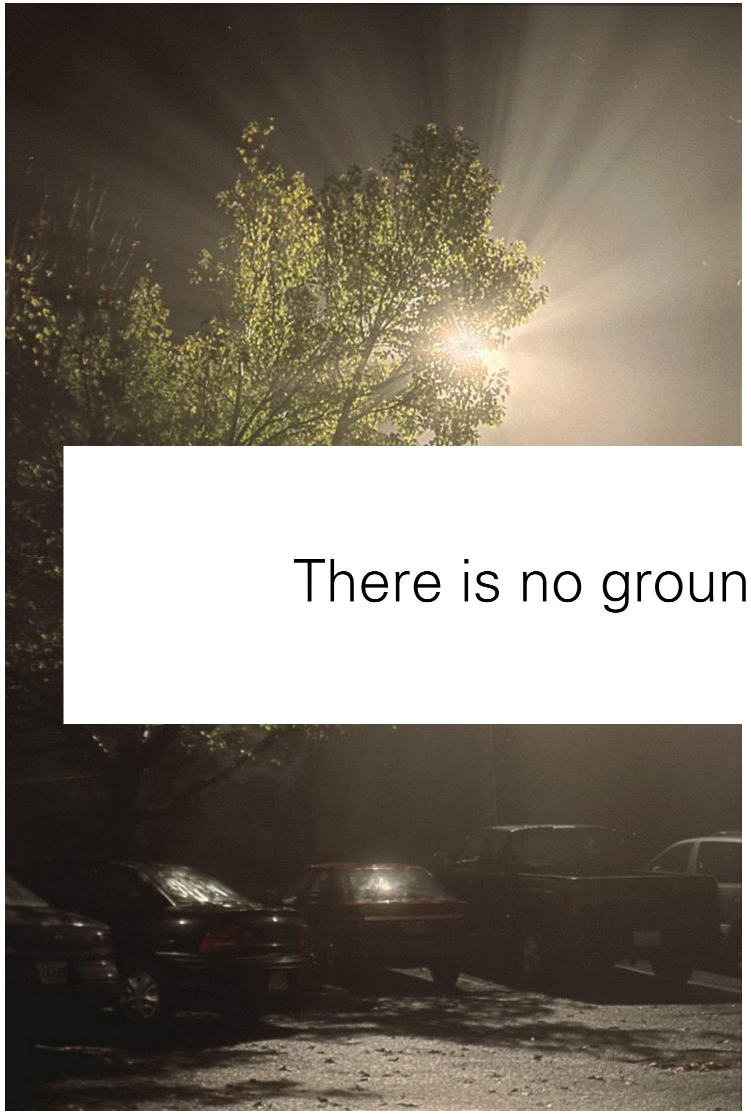


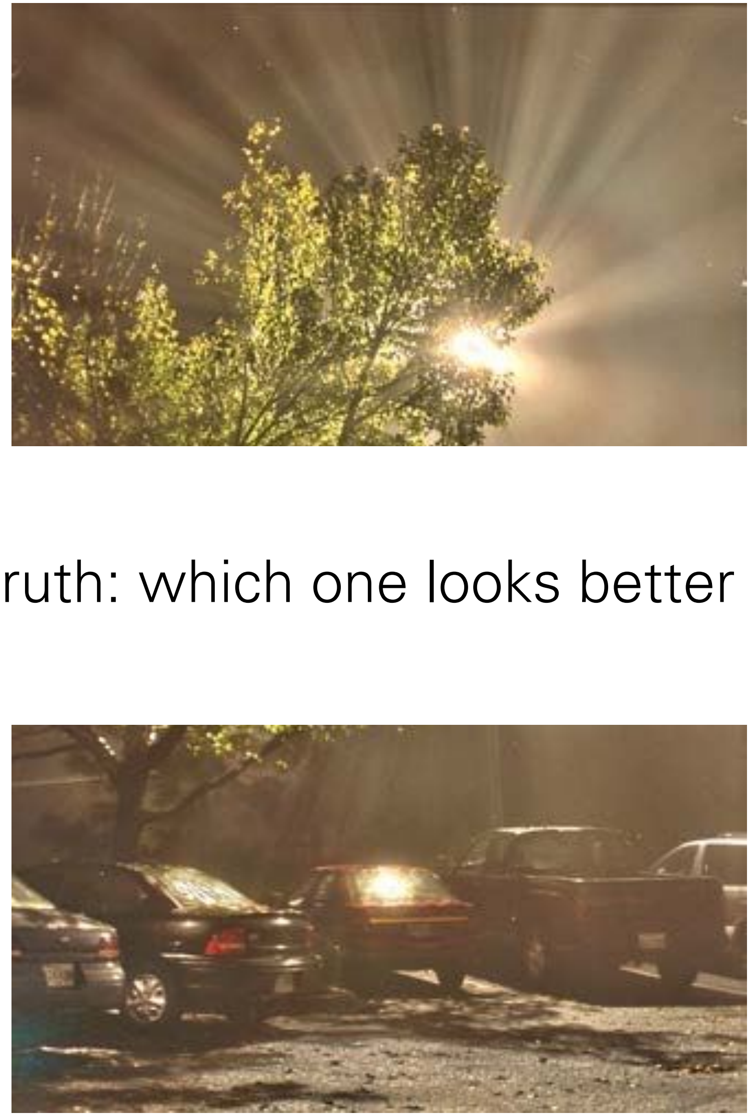
photographic

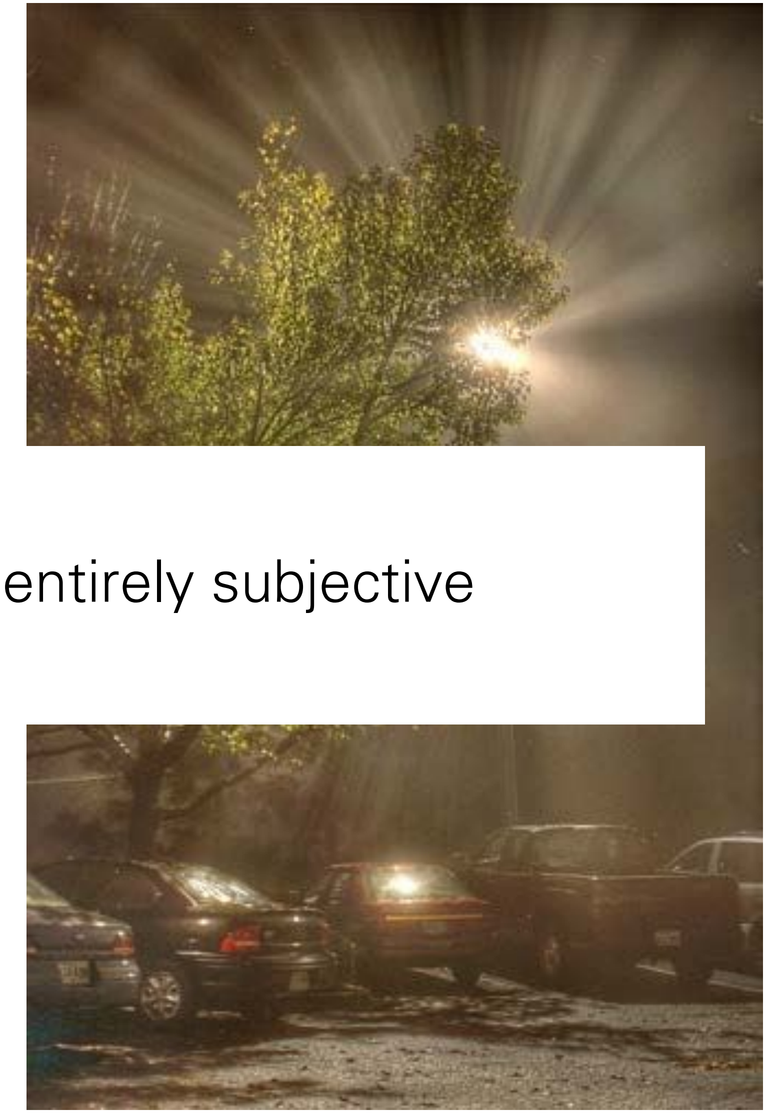
bilateral filtering

gradient-domain

Comparison (which one do you like better?)







There is no ground-truth: which one looks better is entirely subjective

photographic

bilateral filtering

gradient-domain

Tonemapping for a single image

Modern DSLR sensors capture about 3 stops of dynamic range.

- Tonemap single RAW file instead of using camera's default rendering.

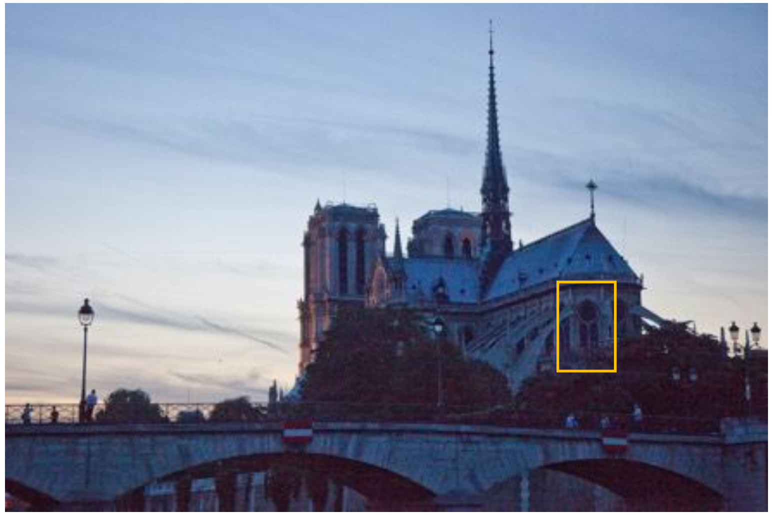
result from image processing pipeline (basic tone reproduction)

tonemapping using bilateral filtering (I think)

Tonemapping for a single image

Modern DSLR sensors capture about 3 stops of dynamic range.

- Tonemap single RAW file instead of using camera’s default rendering.



Careful not to “tonemap” noise.

Some notes about HDR imaging and tonemapping

Our devices do not match the real world

• 10:1	photographic print (higher for glossy paper)
• 20:1	artist's paints
• 200:1	slide film
• 500:1	negative film
• 1000:1	LCD display
• 2000:1	digital SLR (at 12 bits)
• 100000:1	real world

HDR imaging and tonemapping are distinct techniques with different goals

Two challenges:

HDR imaging compensates for sensor limitations

1. HDR imaging – which parts of the world do we measure in the 8-14 bits available to our sensor?
2. Tonemapping – which parts of the world do we show in the 4-10 bits available to our display?

Tonemapping compensates for display limitations

A note about terminology

"High-dynamic-range imaging" is used to refer to a lot of different things:

1. Using single RAW images.
2. Performing radiometric calibration.
3. Merging an exposure stack.
4. Tonemapping an image (linear or non-linear, HDR or LDR).
5. Some or all of the above.

Technically, HDR imaging and tonemapping are distinct processes:

- HDR imaging is the process of creating a radiometrically linear image, free of overexposure and underexposure artifacts. This is achieved using some combination of 1-3, depending on the imaging scenario.
- Tonemapping (step 4) process of mapping the intensity values in an image (linear or non-linear, HDR or LDR) to the range of tones available in a display.

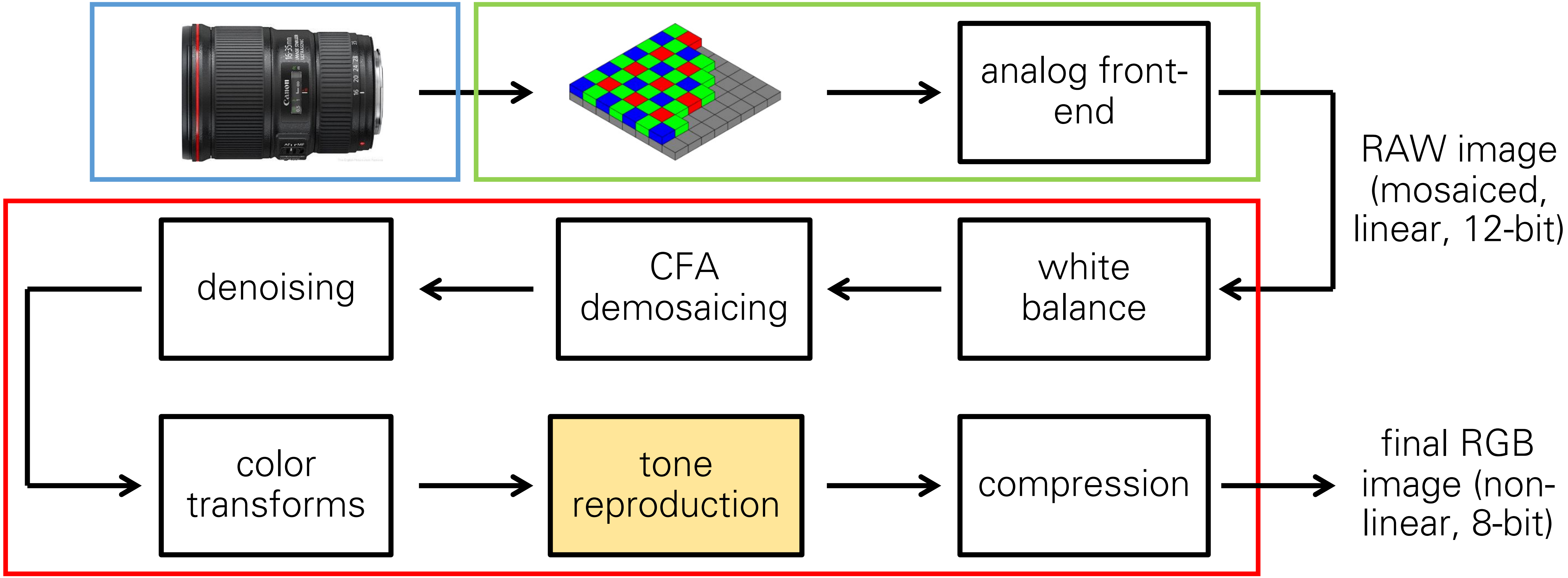
But:

- In consumer photography, "HDR photography" is often used to refer to both HDR imaging (steps 1-3) and tonemapping (step 4).

Another note about terminology

Tonemapping is just another form of tone reproduction.

- Many ISPs implement the tonemapping algorithms we discussed for tone reproduction.



A note of caution

- HDR photography can produce very visually compelling results.

A note of caution

- HDR photography can produce very visually compelling results.

- It is also a very routinely abused technique, resulting in awful results.

A note of caution

- HDR photography can produce very visually compelling results.
- It is also a very routinely abused technique, resulting in awful results.
- The problem typically is tonemapping, not HDR imaging itself.

A note about HDR today

- Most cameras (even phone cameras) have automatic HDR modes/apps.

- Popular-enough feature that phone manufacturers are actively competing about which one has the best HDR.

- The technology behind some of those apps (e.g., Google’s HDR+) is published in SIGGRAPH and SIGGRAPH Asia conferences.

Burst photography for high dynamic range and low-light imaging on mobile cameras

Samuel W. Hasinoff, Jonathan T. Barron, Dillon Sharlet, Florian Kainz, Ryan Geiss, Jiawen Chen, Andrew Adams, Marc Levoy
Google Research

Figure 1: *A comparison of a conventional camera pipeline (left, middle) and our burst photography pipeline (right) running on the same cell-phone camera. In this low-light setting (about 0.7 lux), the conventional camera pipeline underexposes (left). Brightening the image (middle) reveals heavy spatial denoising, which results in loss of detail and an unpleasantly blotchy appearance. Fusing a burst of images increases the signal-to-noise ratio, making aggressive spatial denoising unnecessary. We encourage the reader to zoom in. While our pipeline excels in low-light and high-dynamic-range scenes (for an example of the latter see figure 10), it is computationally efficient and reliably artifact-free, so it can be deployed on a mobile camera and used as a substitute for the conventional pipeline in almost all circumstances. For readability the figure has been made uniformly brighter than the original photographs.*

Abstract

Cell phone cameras have small apertures, which limits the number of photons they can gather, leading to noisy images in low light. They also have small sensor pixels, which limits the number of electrons each pixel can store, leading to limited dynamic range. We describe a computational photography pipeline that captures, aligns, and merges a burst of frames to reduce noise and increase dynamic range. Our system has several key features that help make it robust and efficient. First, we do not use bracketed exposures. Instead, we capture frames of constant exposure, which makes alignment more robust, and we set this exposure low enough to avoid blowing out highlights. The resulting merged image has clean shadows and high bit depth, allowing us to apply standard HDR tone mapping methods. Second, we begin from Bayer raw frames rather than the demosaicked RGB (or YUV) frames produced by hardware Image Signal Processors (ISPs) common on mobile platforms. This gives us more bits per pixel and allows us to circumvent the ISP’s unwanted tone mapping and spatial denoising. Third, we use a novel FFT-based alignment algorithm and a hybrid 2D/3D Wiener filter to denoise and merge the frames in a burst. Our implementation is built atop Android’s Camera2 API, which provides per-frame camera control and access to raw imagery, and is written in the Halide domain-specific language (DSL). It runs in 4 seconds on device (for a 12 Mpix image), requires no user intervention, and ships on several mass-produced cell phones.

Keywords: computational photography, high dynamic range

Concepts: •**Computing methodologies → Computational photography; Image processing;**

1 Introduction

The main technical impediment to better photographs is lack of light. In indoor or night-time shots, the scene as a whole may provide insufficient light. The standard solution is either to apply analog or digital gain, which amplifies noise, or to lengthen exposure time, which causes motion blur due to camera shake or subject motion. Surprisingly, daytime shots with high dynamic range may also suffer from lack of light. In particular, if exposure time is reduced to avoid

SA '16 Technical Papers, December 05 - 08, 2016, Macao
ISBN: 978-1-4503-4514-9/16/12
DOI: http://dx.doi.org/10.1145/2980179.2980254

Optimal weights for HDR merging

Merging non-linear exposure stacks

1. Calibrate response curve
2. Linearize images

For each pixel:

3. Find “valid” images ← (noise) $0.05 <$ pixel < 0.95 (clipping)
4. Weight valid pixel values appropriately ← (pixel value) / t_i
5. Form a new pixel value as the weighted average of valid pixel values

→ Same steps as in the RAW case.

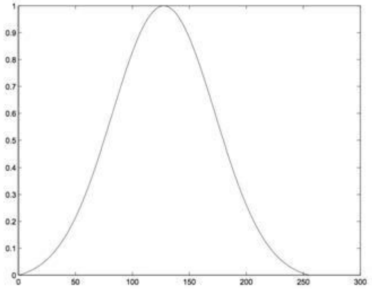
Note: many possible weighting schemes

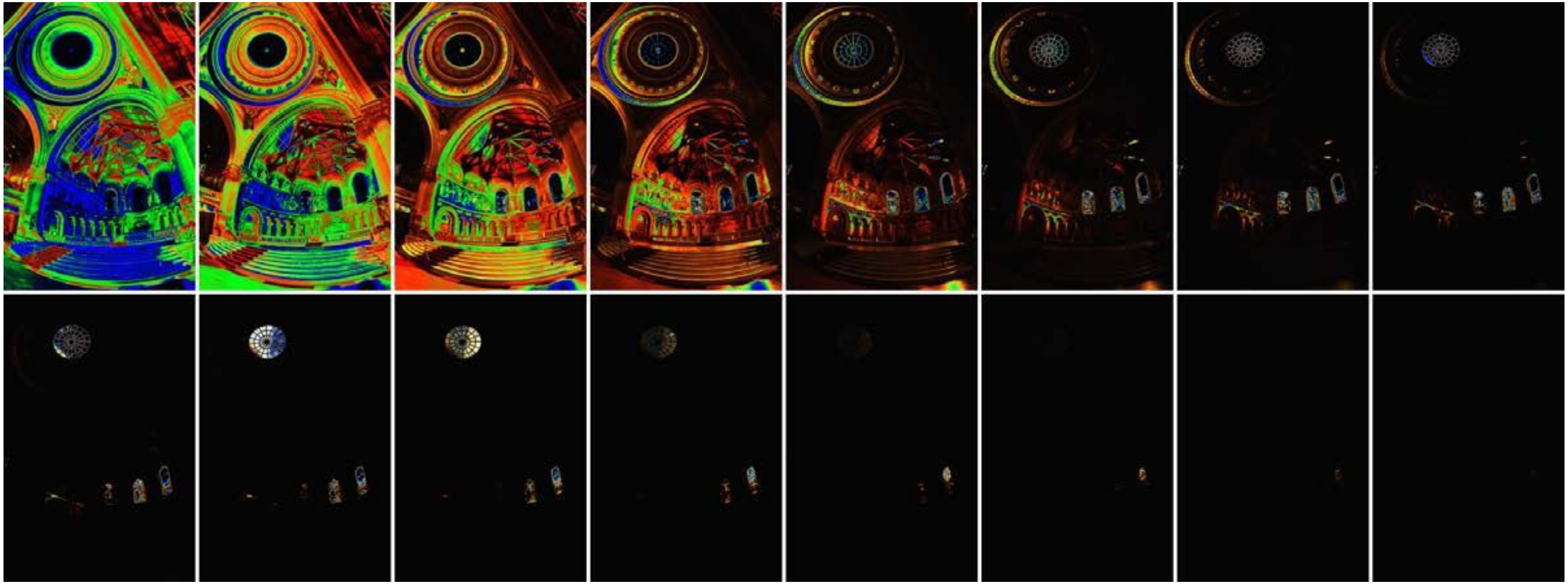
Many possible weighting schemes

- What are the optimal weights for merging an exposure stack?

"Confidence" that pixel is noisy/clipped

$$w_{ij} = \exp\left(-4\frac{\left(I_{lin_{ij}} - 0.5\right)^2}{0.5^2}\right)$$





RAW (linear) image formation model

(Weighted) radiant flux for image pixel (x,y): $\alpha \cdot \Phi(x, y)$

Exposure time:

t_5 t_4 t_3 t_2 t_1

What weights should we use to merge these images, so that the resulting HDR image is an optimal estimator of the weighted radiant flux?

Different images in the exposure stack will have different noise characteristics

Simple estimation example

We have two independent unbiased estimators x and y of the same quantity I (e.g., pixel intensity) with variance $\sigma[\mathrm{x}]^2$ and $\sigma[\mathrm{y}]^2$.

What does unbiased mean?

Simple estimation example

We have two independent unbiased estimators x and y of the same quantity I (e.g., pixel intensity) with variance $\sigma[\mathrm{x}]^2$ and $\sigma[\mathrm{y}]^2$.

What does unbiased mean?

$$\mathrm{E}[\mathrm{x}] = \mathrm{E}[\mathrm{y}] = \mathrm{I}$$

Assume we form a new estimator from the convex combination of the other two:

$$\mathrm{z} = \mathrm{a} \cdot \mathrm{x} + (1 - \mathrm{a}) \cdot \mathrm{y}$$

Is the new estimator z unbiased?

Simple estimation example

We have two independent unbiased estimators x and y of the same quantity I (e.g., pixel intensity) with variance $\sigma[x]^2$ and $\sigma[y]^2$.

What does unbiased mean?

$$E[x] = E[y] = I$$

Assume we form a new estimator from the convex combination of the other two:

$$z = a \cdot x + (1 - a) \cdot y$$

Is the new estimator z unbiased? → Yes, convex combination preserves unbiasedness.

$$E[z] = E[a \cdot x + (1 - a) \cdot y] = a \cdot E[x] + (1 - a) \cdot E[y] = I$$

How should we select a?

Simple estimation example

We have two independent unbiased estimators x and y of the same quantity I (e.g., pixel intensity) with variance $\sigma[x]^2$ and $\sigma[y]^2$.

What does unbiased mean?

$$E[x] = E[y] = I$$

Assume we form a new estimator from the convex combination of the other two:

$$z = a \cdot x + (1 - a) \cdot y$$

Is the new estimator z unbiased? → Yes, convex combination preserves unbiasedness.

$$E[z] = E[a \cdot x + (1 - a) \cdot y] = a \cdot E[x] + (1 - a) \cdot E[y] = I$$

How should we select a? → Minimize variance (= expected squared error for unbiased estimators).

$$E[(z - I)^2] = E[z^2] - 2 \cdot E[z] \cdot I + I^2 = E[z^2] - E[z]^2 = \sigma[z]^2$$

What is the variance of z as a function of a?

Simple estimation example

We have two independent unbiased estimators x and y of the same quantity I (e.g., pixel intensity) with variance $\sigma[\mathrm{x}]^2$ and $\sigma[\mathrm{y}]^2$.

What does unbiased mean?

$$\mathrm{E}[\mathrm{x}] = \mathrm{E}[\mathrm{y}] = \mathrm{I}$$

Assume we form a new estimator from the convex combination of the other two:

$$\mathrm{z} = \mathrm{a} \cdot \mathrm{x} + (1 - \mathrm{a}) \cdot \mathrm{y}$$

Is the new estimator z unbiased? → Yes, convex combination preserves unbiasedness.

$$\mathrm{E}[\mathrm{z}] = \mathrm{E}[\mathrm{a} \cdot \mathrm{x} + (1 - \mathrm{a}) \cdot \mathrm{y}] = \mathrm{a} \cdot \mathrm{E}[\mathrm{x}] + (1 - \mathrm{a}) \cdot \mathrm{E}[\mathrm{y}] = \mathrm{I}$$

How should we select a? → Minimize variance (= expected squared error for unbiased estimators).

$$\mathrm{E}[(\mathrm{z} - \mathrm{I})^2] = \mathrm{E}[\mathrm{z}^2] - 2 \cdot \mathrm{E}[\mathrm{z}] \cdot \mathrm{I} + \mathrm{I}^2 = \mathrm{E}[\mathrm{z}^2] - \mathrm{E}[\mathrm{z}]^2 = \sigma[\mathrm{z}]^2$$

What is the variance of z as a function of a?

$$\sigma[\mathrm{z}]^2 = \mathrm{a}^2 \cdot \sigma[\mathrm{x}]^2 + (1 - \mathrm{a})^2 \cdot \sigma[\mathrm{y}]^2$$

What value of a minimizes $\sigma[\mathrm{z}]^2$?

Simple estimation example

Simple optimization problem:

$$\frac{\partial \sigma[z]^2}{\partial a} = 0$$

$$\Rightarrow \quad \frac{\partial (a^2 \cdot \sigma[x]2 + (1 - a)^2 \cdot \sigma[y]^2)}{\partial a} = 0$$

$$\Rightarrow \quad 2 \cdot a \cdot \sigma[x]2 - 2 \cdot (1 - a) \cdot \sigma[y]^2 = 0$$

$$\Rightarrow \quad a = \frac{\sigma[y]^2}{\sigma[x]^2 + \sigma[y]^2} \qquad \text{and} \qquad 1 - a = \frac{\sigma[x]^2}{\sigma[x]^2 + \sigma[y]^2}$$

Simple estimation example

Putting it all together, the optimal linear combination of the two estimators is

$$z = \underbrace{\frac{\sigma[x]^2\sigma[y]2}{\sigma[x]^2 + \sigma[y]2}}_{\substack{\text{normalization} \\ \text{factor}}} \cdot \underbrace{\left(\frac{1}{\sigma[x]2}x + \frac{1}{\sigma[y]^2}y\right)}_{\substack{\text{weights inversely} \\ \text{proportional to variance}}}$$

Simple estimation example

Putting it all together, the optimal linear combination of the two estimators is

$$z = \underbrace{\frac{\sigma[\mathrm{x}]^2\sigma[\mathrm{y}]2}{\sigma[\mathrm{x}]^2 + \sigma[\mathrm{y}]2}}_{\substack{\text{normalization}\\ \text{factor}}} \cdot \underbrace{\left(\frac{1}{\sigma[\mathrm{x}]2}x + \frac{1}{\sigma[\mathrm{y}]^2}y\right)}_{\substack{\text{weights inversely}\\ \text{proportional to variance}}}$$

More generally, for more than two estimators,

$$z = \frac{1}{\sum_{i=1}^{N}\frac{1}{\sigma[x_i]^2}} \cdot \sum_{i=1}^{N}\frac{1}{\sigma[x_i]^2}x_i$$

This is weighting scheme is called Fisher weighting and is a BLUE estimator.

Back to HDR

Given unclipped and dark-frame-corrected intensity measurements $\mathrm{I}_{\mathrm{i}}[\mathrm{x}, \mathrm{y}]$ at pixel $[\mathrm{x}, \mathrm{y}]$ and exposures t_{i}, we can merge them optimally into a single HDR intensity $\mathrm{I}[\mathrm{x}, \mathrm{y}]$ as

$$I[x, y] = \frac{1}{\sum_{i=1}^{N} w_i\,[x, y]} \cdot \sum_{i=1}^{N} w_i[x, y] \frac{1}{t_{\mathrm{i}}} I_{i[x, y]}$$

Back to HDR

Given unclipped and dark-frame-corrected intensity measurements $\mathrm{I_i}[\mathrm{x}, \mathrm{y}]$ at pixel $[\mathrm{x}, \mathrm{y}]$ and exposures $\mathrm{t_i}$, we can merge them optimally into a single HDR intensity $\mathrm{I}[\mathrm{x}, \mathrm{y}]$ as

$$I[x,y] = \frac{1}{\sum_{i=1}^{N} w_i\,[x,y]} \cdot \sum_{i=1}^{N} w_i[x,y]\,\frac{1}{t_\mathrm{i}} I_{i[x,y]} = \frac{1}{\sum_{i=1}^{N} \frac{1}{\sigma[\frac{1}{t_\mathrm{i}} I_i[x,y]]^2}} \cdot \sum_{i=1}^{N} \frac{1}{\sigma[\frac{1}{t_\mathrm{i}} I_i[x,y]]^2} \frac{1}{t_\mathrm{i}} I_{i[x,y]}$$

The <u>per-pixel</u> weights $\mathrm{w_i}[\mathrm{x}, \mathrm{y}]$ should be selected to be inversely proportional to the variance $\sigma[\frac{1}{\mathrm{t_i}} \mathrm{I_i}[\mathrm{x}, \mathrm{y}]]^2$ at each image in the exposure stack.

- How do we compute this variance?

Pixel noise and variance

- Recall: noise is characterized by its variance
  - i.e. each pixel value comes from a true value plus some noise added
- We can calibrate this noise by taking multiple exposures, or we can derive variance equations using pen and paper

Sources of noise

- Photon noise
  - variance proportional to signal
  - dominates for dark pixels
- Read noise
  - constant variance
  - dominates for dark pixels
- Affine noise model: $I = L \cdot g + n_{\text{add}}$ where $n_{\text{add}} = n_{\text{read}} \cdot g + n_{\text{ADC}}$
- For a pixel value I : $\sigma(I)^2 = t \cdot (a \cdot \Phi + D) \cdot g^2 + \sigma_{\text{add}}^2$
  - where $\sigma_{\text{add}}^2 = \sigma_{\text{read}}^2 \cdot g^2 + \sigma_{\text{ADC}}^2$, a and σ_{read}^2 depend on the camera and ISO

Back to HDR

Given unclipped and dark-frame-corrected intensity measurements $\mathrm{I_i}[\mathrm{x},\mathrm{y}]$ at pixel $[\mathrm{x},\mathrm{y}]$ and exposures $\mathrm{t_i}$, we can merge them optimally into a single HDR intensity $\mathrm{I}[\mathrm{x},\mathrm{y}]$ as

$$I[x,y] = \frac{1}{\sum_{i=1}^{N} w_i\,[x,y]} \cdot \sum_{i=1}^{N} w_i[x,y]\,\frac{1}{t_\mathrm{i}} I_i[x,y] = \frac{1}{\sum_{i=1}^{N} \frac{1}{\sigma[\frac{1}{t_\mathrm{i}} I_i[x,y]]^2}} \cdot \sum_{i=1}^{N} \frac{1}{\sigma[\frac{1}{t_\mathrm{i}} I_i[x,y]]^2} \frac{1}{t_\mathrm{i}} I_i[x,y]$$

The <u>per-pixel</u> weights $\mathrm{w_i}[\mathrm{x},\mathrm{y}]$ should be selected to be inversely proportional to the variance $\sigma[\frac{1}{\mathrm{t_i}}\mathrm{I_i}[\mathrm{x},\mathrm{y}]]^2$ at each image in the exposure stack.

- How do we compute this variance?

Back to HDR

Given unclipped and dark-frame-corrected intensity measurements $\mathrm{I_i}[\mathrm{x}, \mathrm{y}]$ at pixel $[\mathrm{x}, \mathrm{y}]$ and exposures $\mathrm{t_i}$, we can merge them optimally into a single HDR intensity $\mathrm{I}[\mathrm{x}, \mathrm{y}]$ as

$$I[x,y] = \frac{1}{\sum_{i=1}^{N} w_i\,[x,y]} \cdot \sum_{i=1}^{N} w_i[x,y]\,\frac{1}{t_\mathrm{i}} I_i[x,y] = \frac{1}{\sum_{i=1}^{N} \frac{1}{\sigma[\frac{1}{t_\mathrm{i}} I_i[x,y]]^2}} \cdot \sum_{i=1}^{N} \frac{1}{\sigma[\frac{1}{t_\mathrm{i}} I_i[x,y]]^2}\,\frac{1}{t_\mathrm{i}} I_i[x,y]$$

The <u>per-pixel</u> weights $\mathrm{w_i}[\mathrm{x}, \mathrm{y}]$ should be selected to be inversely proportional to the variance $\sigma[\frac{1}{\mathrm{t_i}} \mathrm{I_i}[\mathrm{x}, \mathrm{y}]]^2$ at each image in the exposure stack.

- How do we compute this variance? $\rightarrow$ Use affine noise model.

$$\sigma[\frac{1}{t_\mathrm{i}} I_i[x,y]]^2 = ?$$

Back to HDR

Given unclipped and dark-frame-corrected intensity measurements $\mathrm{I_i}[\mathrm{x},\mathrm{y}]$ at pixel $[\mathrm{x},\mathrm{y}]$ and exposures $\mathrm{t_i}$, we can merge them optimally into a single HDR intensity $\mathrm{I}[\mathrm{x},\mathrm{y}]$ as

$$I[x,y] = \frac{1}{\sum_{i=1}^{N} w_i\,[x,y]} \cdot \sum_{i=1}^{N} w_i[x,y]\,\frac{1}{t_\mathrm{i}}\,I_i[x,y] = \frac{1}{\sum_{i=1}^{N} \frac{1}{\sigma[\frac{1}{t_\mathrm{i}} I_i[x,y]]^2}} \cdot \sum_{i=1}^{N} \frac{1}{\sigma[\frac{1}{t_\mathrm{i}} I_i[x,y]]^2}\,\frac{1}{t_\mathrm{i}}\,I_i[x,y]$$

The <u>per-pixel</u> weights $\mathrm{w_i}[\mathrm{x},\mathrm{y}]$ should be selected to be inversely proportional to the variance $\sigma[\frac{1}{\mathrm{t_i}}\mathrm{I_i}[\mathrm{x},\mathrm{y}]]^2$ at each image in the exposure stack.

- How do we compute this variance? $\rightarrow$ Use affine noise model.

$$\sigma[\frac{1}{t_\mathrm{i}} I_i[x,y]]^2 = \frac{1}{t_\mathrm{i}^2}\,\sigma[I_i[x,y]]^2$$

$$\Rightarrow \quad \sigma[\frac{1}{t_\mathrm{i}} I_i[x,y]]^2 = ?$$

Back to HDR

Given unclipped and dark-frame-corrected intensity measurements $\mathrm{I_i}[\mathrm{x}, \mathrm{y}]$ at pixel $[\mathrm{x}, \mathrm{y}]$ and exposures $\mathrm{t_i}$, we can merge them optimally into a single HDR intensity $\mathrm{I}[\mathrm{x}, \mathrm{y}]$ as

$$I[x,y] = \frac{1}{\sum_{i=1}^{N} w_i\,[x,y]} \cdot \sum_{i=1}^{N} w_i[x,y]\,\frac{1}{t_\mathrm{i}} I_{i}[x,y] = \frac{1}{\sum_{i=1}^{N} \frac{1}{\sigma[\frac{1}{t_\mathrm{i}} I_i[x,y]]^2}} \cdot \sum_{i=1}^{N} \frac{1}{\sigma[\frac{1}{t_\mathrm{i}} I_i[x,y]]^2} \frac{1}{t_\mathrm{i}} I_{i}[x,y]$$

The <u>per-pixel</u> weights $\mathrm{w_i}[\mathrm{x}, \mathrm{y}]$ should be selected to be inversely proportional to the variance $\sigma[\frac{1}{\mathrm{t_i}} \mathrm{I_i}[\mathrm{x}, \mathrm{y}]]^2$ at each image in the exposure stack.

- How do we compute this variance? → Use affine noise model.

$$\sigma[\frac{1}{t_\mathrm{i}} I_i[x,y]]^2 = \frac{1}{t_\mathrm{i}^2}\,\sigma[I_i[x,y]]^2$$

$$\Rightarrow \quad \sigma[\frac{1}{t_\mathrm{i}} I_i[x,y]]^2 = \frac{1}{t_\mathrm{i}^2}\,(t_\mathrm{i} \cdot \alpha \cdot \Phi[x,y] \cdot g^2 + \sigma_\mathrm{add}^2)$$

Computing the optimal weights requires:

1. calibrated noise characteristics.
2. knowing the radiant flux $\alpha \cdot \Phi[\mathrm{x}, \mathrm{y}]$.

Back to HDR

Given unclipped and dark-frame-corrected intensity measurements $\mathrm{I_i}[\mathrm{x},\mathrm{y}]$ at pixel $[\mathrm{x},\mathrm{y}]$ and exposures $\mathrm{t_i}$, we can merge them optimally into a single HDR intensity $\mathrm{I}[\mathrm{x},\mathrm{y}]$ as

$$I[x,y] = \frac{1}{\sum_{i=1}^{N} w_i\,[x,y]} \cdot \sum_{i=1}^{N} w_i[x,y] \frac{1}{t_\mathrm{i}} I_i[x,y] = \frac{1}{\sum_{i=1}^{N} \frac{1}{\sigma[\frac{1}{t_\mathrm{i}} I_i[x,y]]^2}} \cdot \sum_{i=1}^{N} \frac{1}{\sigma[\frac{1}{t_\mathrm{i}} I_i[x,y]]^2} \frac{1}{t_\mathrm{i}} I_i[x,y]$$

The <u>per-pixel</u> weights $\mathrm{w_i}[\mathrm{x},\mathrm{y}]$ should be selected to be inversely proportional to the variance $\sigma[\frac{1}{\mathrm{t_i}}\mathrm{I_i}[\mathrm{x},\mathrm{y}]]^2$ at each image in the exposure stack.

- How do we compute this variance? → Use affine noise model.

$$\sigma[\frac{1}{t_\mathrm{i}} I_i[x,y]]^2 = \frac{1}{t_\mathrm{i}^2}\sigma[I_i[x,y]]^2$$

$$\Rightarrow \quad \sigma[\frac{1}{t_\mathrm{i}} I_i[x,y]]^2 = \frac{1}{t_\mathrm{i}^2}(t_\mathrm{i} \cdot \alpha \cdot \Phi[x,y] \cdot g^2 + \sigma_\mathrm{add}{}^2)$$

Computing the optimal weights requires:

1. calibrated noise characteristics.
2. $\boxed{\text{knowing the radiant flux } \alpha \cdot \Phi[\mathrm{x},\mathrm{y}].}$

This is what we wanted to estimate!

Simplification: only photon noise

If we assume that our measurements are dominated by photon noise, the variance becomes:

$$\sigma[\frac{1}{t_\mathrm{i}} I_i[x,y]]^2 = \frac{1}{{t_\mathrm{i}}^2}\,(t_\mathrm{i} \cdot \alpha \cdot \Phi[x,y] \cdot g^2 + {\sigma_\mathrm{add}}^2) \simeq ?$$

Simplification: only photon noise

If we assume that our measurements are dominated by photon noise, the variance becomes:

$$\sigma[\frac{1}{t_{\mathrm{i}}} I_i[x,y]]^2 = \frac{1}{{t_{\mathrm{i}}}^2}\left(t_{\mathrm{i}} \cdot \alpha \cdot \Phi[x,y] \cdot g^2 + {\sigma_{\mathrm{add}}}^2\right) \simeq \frac{1}{t_{\mathrm{i}}} \alpha \cdot \Phi[x,y] \cdot g^2$$

By replacing in the merging formula and assuming only valid pixels, the HDR estimate becomes:

$$I[x,y] = \frac{1}{\sum_{i=1}^{N} \frac{1}{\frac{1}{t_{\mathrm{i}}} \alpha \cdot \Phi[x,y] \cdot g^2}} \cdot \sum_{i=1}^{N} \frac{1}{\frac{1}{t_{\mathrm{i}}} \alpha \cdot \Phi[x,y] \cdot g^2} \frac{1}{t_{\mathrm{i}}} I_{i[x,y]}$$

Simplification: only photon noise

If we assume that our measurements are dominated by photon noise, the variance becomes:

$$\sigma\left[\frac{1}{t_\mathrm{i}} I_i[x,y]\right]^2 = \frac{1}{t_\mathrm{i}^2}\left(t_\mathrm{i} \cdot \alpha \cdot \Phi[x,y] \cdot g^2 + \sigma_{\mathrm{add}}^2\right) \simeq \frac{1}{t_\mathrm{i}} \alpha \cdot \Phi[x,y] \cdot g^2$$

By replacing in the merging formula and assuming only valid pixels, the HDR estimate becomes:

$$I[x,y] = \frac{1}{\sum_{i=1}^{N} \frac{1}{\frac{1}{t_\mathrm{i}} \cancel{\alpha \cdot \Phi[x,y] \cdot g^2}}} \cdot \sum_{i=1}^{N} \frac{1}{\cancel{\frac{1}{t_\mathrm{i}}} \cancel{\alpha \cdot \Phi[x,y] \cdot g^2}} \cancel{\frac{1}{t_\mathrm{i}}} I_i[x,y] = \frac{1}{\sum_{i=1}^{N} t_i} \cdot \sum_{i=1}^{N} I_i[x,y]$$

Simplification: only photon noise

If we assume that our measurements are dominated by photon noise, the variance becomes:

$$\sigma[\frac{1}{t_\mathrm{i}} I_i[x,y]]^2 = \frac{1}{{t_\mathrm{i}}^2}(t_\mathrm{i} \cdot \alpha \cdot \Phi[x,y] \cdot g^2 + {\sigma_{\mathrm{add}}}^2) \simeq \frac{1}{t_\mathrm{i}} \alpha \cdot \Phi[x,y] \cdot g^2$$

By replacing in the merging formula and assuming only valid pixels, the HDR estimate becomes:

$$I[x,y] = \frac{1}{\sum_{i=1}^{N} \frac{1}{\frac{1}{t_\mathrm{i}} \cancel{\alpha \cdot \Phi[x,y] \cdot g^2}}} \cdot \sum_{i=1}^{N} \frac{1}{\cancel{\frac{1}{t_\mathrm{i}}} \cancel{\alpha \cdot \Phi[x,y] \cdot g^2}} \cancel{\frac{1}{t_\mathrm{i}}} I_i[x,y] = \boxed{\frac{1}{\sum_{i=1}^{N} t_i} \cdot \sum_{i=1}^{N} I_i[x,y]}$$

Notice that we no longer weight each image in the exposure stack by its exposure time!

Some comparisons

original weights

optimal weights assuming
only photon noise

More general case

If we cannot assume that our measurements are dominated by photon noise, we can approximate the variance as:

$$\sigma[\frac{1}{t_{\mathrm{i}}} I_i[x,y]]^2 = \frac{1}{t_{\mathrm{i}}{}^2}\,(t_{\mathrm{i}} \cdot \alpha \cdot \Phi[x,y] \cdot g^2 + \sigma_{\mathrm{add}}{}^2) \simeq \frac{1}{t_{\mathrm{i}}{}^2}\,(I_i[x,y] \cdot g + \sigma_{\mathrm{add}}{}^2)$$

Where does this approximation come from?

More general case

If we cannot assume that our measurements are dominated by photon noise, we can approximate the variance as:

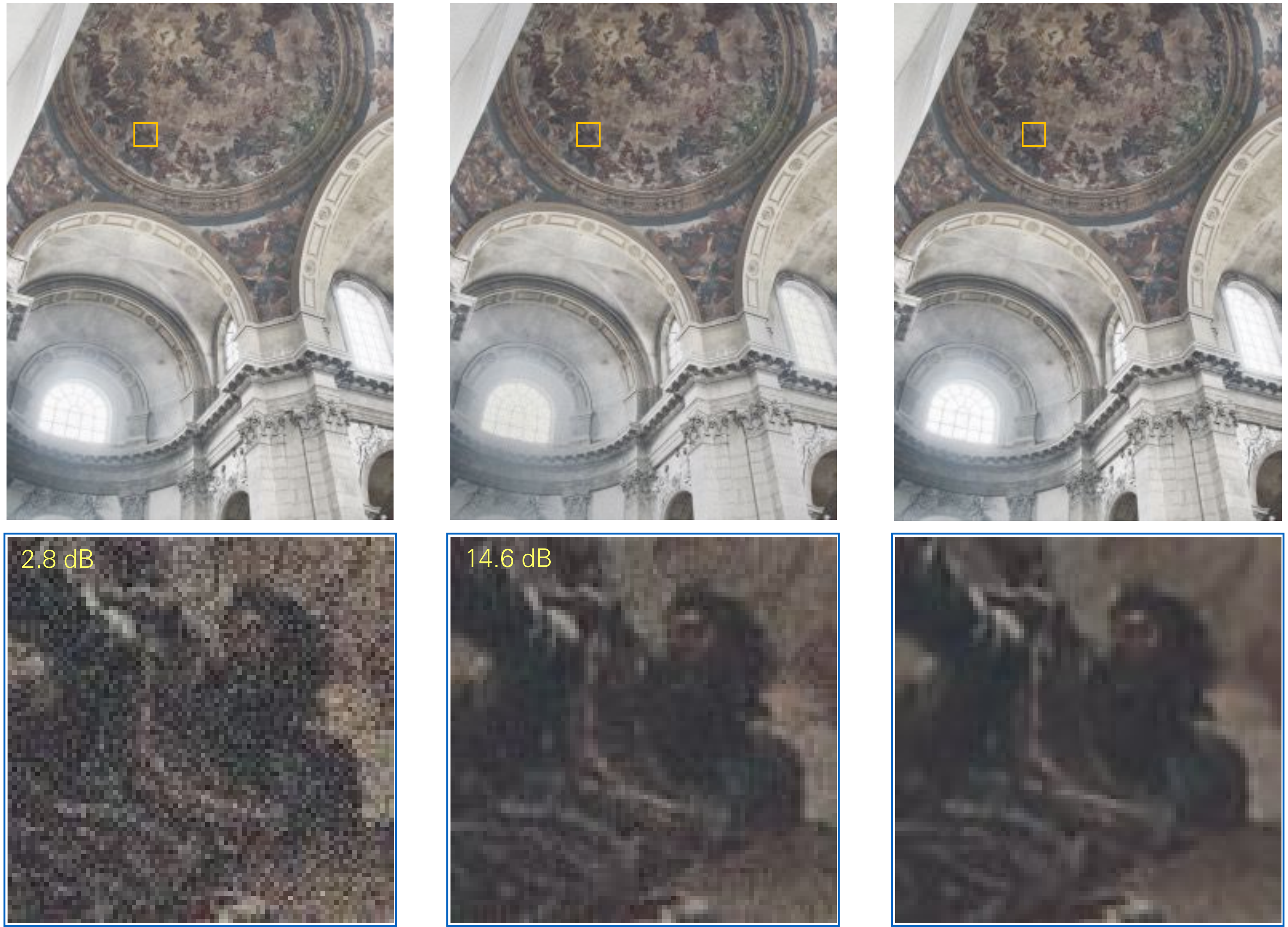
$$\sigma[\frac{1}{t_{\mathrm{i}}} I_i[x,y]]^2 = \frac{1}{t_{\mathrm{i}}^2}\left(t_{\mathrm{i}} \cdot \alpha \cdot \Phi[x,y] \cdot g^2 + \sigma_{\mathrm{add}}^2\right) \simeq \frac{1}{t_{\mathrm{i}}^2}\left(I_i[x,y] \cdot g + \sigma_{\mathrm{add}}^2\right)$$

Where does this approximation come from?

- We use the fact that each pixel intensity (after dark frame subtraction) is an unbiased estimate of the radiant flux, weighted by exposure and gain:

$$\mathrm{E}[I_i[x,y]] = t_{\mathrm{i}} \cdot \alpha \cdot \Phi[x,y] \cdot \mathrm{g}$$

Some comparisons



What about ISO?

Noise-Optimal Capture for High Dynamic Range Photography

Samuel W. Hasinoff　Frédo Durand　William T. Freeman
Massachusetts Institute of Technology
Computer Science and Artificial Intelligence Laboratory

Abstract

Taking multiple exposures is a well-established approach both for capturing high dynamic range (HDR) scenes and for noise reduction. But what is the optimal *set of photos to capture? The typical approach to HDR capture uses a set of photos with geometrically-spaced exposure times, at a fixed ISO setting (typically ISO 100 or 200). By contrast, we show that the capture sequence with optimal worst-case performance, in general, uses much higher and variable ISO settings, and spends longer capturing the dark parts of the scene. Based on a detailed model of noise, we show that optimal capture can be formulated as a mixed integer programming problem. Compared to typical HDR capture, our method lets us achieve higher worst-case SNR in the same capture time (for some cameras, up to* 19 dB *improvement in the darkest regions), or much faster capture for the same minimum acceptable level of SNR. Our experiments demonstrate this advantage for both real and synthetic scenes.*

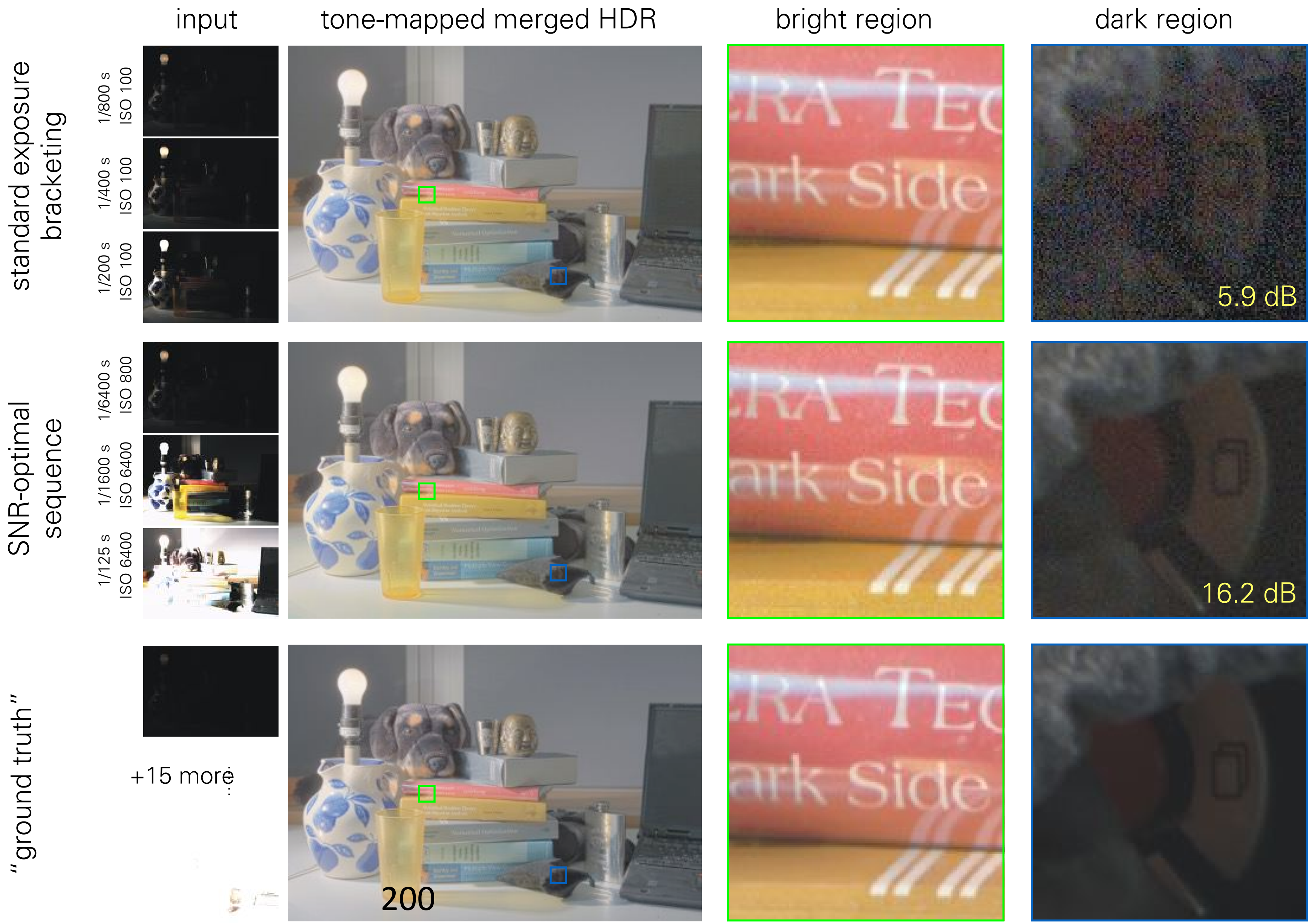
rameters of an exposure sequence, and we show that this reduces to solving a mixed integer programming problem. In particular, we show that, contrary to suggested practice (*e.g.*, [5]), using high ISO values is desirable and can enable significant gains in signal-to-noise ratio.

The most important feature of our noise model is its explicit decomposition of additive noise into pre- and post-amplifier sources (Fig. 1), which constitutes the basis for the high ISO advantage. The same model has been used in several unpublished studies characterizing the noise performance of digital SLR cameras [7, 20], supported by extensive empirical validation. Although all the components in our model are well-established, previous treatments of noise in the vision literature [13, 18] do not model the dependence of noise on ISO setting (*i.e.*, sensor gain).

To the best of our knowledge, varying the ISO setting has not previously been exploited to optimize SNR for high dynamic range capture. However, in the much simpler context of single-shot photography, the *expose to the right* tech-

- We need to separately account for read and ADC noise, as read noise is gain-dependent.
- We can optimize our exposure bracket by varying both shutter speed and ISO

Real capture results



Recap

- High dynamic range (HDR) imaging is useful, and a new aesthetic
  - but is not necessary in all photographic situations
- Low dynamic range (LDR) tone mapping methods can also be applied to HDR scenes
  - but reducing very HDR scenes to 8 bits for JPEG using only global methods is hard
- Local methods reduce large-scale luminance changes (across the image) while preserving local contrast (across edges)
  - use edge-preserving filters to avoid halos

Next Lecture:
Edge-aware filtering,
Gradient-domain image
processing